Portfolio

Roland Lee

(503) 646-4467
rlee@graduates.iti.com

Roland Lee, BSc, AIT

13020 SW Tapadera St. Beaverton, OR 97008
Online portfolio: http://server2.portland.iti.com/rlee

(503) 646-4467
roland_97008@yahoo.com

Objective:

- Seeking an opportunity as a Bionformatics/software engineer in a challenging team-based environment.

Personal Profile:

- Trilingual- fluent in Chinese(Mandarin), English and Filipino(Tagalog).
- 3+ years Quality Assurance experience/ testing/ documentation
- Experience in Visual Basic, Java, DB2, Macromedia Products, HTML, Microsoft Access, SQL, ASP, Java/VBscript.
- Bachelor of Science - Biology Major/ Chemistry Minor
- Continuously strive to remain current in new technologies to enhance ability in application development.
- Enjoy team collaboration towards discovery and problem solving.
- Willing to travel.

IT Skills:

Operating Systems:

- Windows 98/ME/2000
- Windows NT

Development Tools and Technologies:

- Visual Basic 6.0
- Visual Age for Java 3.5
- Java (JDK 1.2, 1.3)
- DB2 UDB
- Microsoft Access
- VBScript/JavaScript
- Visual Basic for Applications
- HTML 4.0
- FrontPage 2000
- ASP
- ADO
- SQL
- COM+/COM
- MTS, IIS

Methodologies:

- GUIDS
- OOD and OOP
- ERD Data Modeling
- UML/RUP
- Design Patterns
- n-tier architecture

Independent Studies:

- Macromedia Flash
- Macromedia Dreamweaver
- Macromedia Fireworks

Current goals:

- CGI/Perl, XML, Javascript and reading about .NET.
- Learn how to implement SQL Server or Oracle database into VB or Java.
- Java Certification

Employment History:

Oligos Etc 1997 - 2000

- Good Manufacturing Processes(GMP) Certified Facility - followed Standard Operating Procedures (SOP's) - stressed quality assurance and documentation of processes in manufacturing of product - required constant communication with multiple departments to ensure proper techniques followed throughout production.
- Cross trained in 3 different departments - quickly learned multiple skills and assisted different departments when needed.
- Managed multiple customer orders with different requirements and ship dates. Excellent time-management skills - able to operate effectively under pressure.
- Involved in purifying experimental strands of DNA for customers with customized requests - resulted in increased satisfaction and customer's willingness for future collaboration. Enjoyed experimentation and team work with fellow colleagues to develop innovative solutions to customer needs.
- Customer oriented quality control - emphasized accuracy and attention to detail especially for clients involved with Food and Drug Administration (FDA) Phase I, II, & III testing resulted in good customer relationships.
- Authored several Standard Operating Procedures(SOP's) to facilitate the implementation of new methods in the laboratory in compliance with the FDA and GMP requirements.

Willamette University Women's Volleyball Assistant Coach 1999 - 2000

- High degree of analysis and strategy, systematic and organized methods of coaching, discussion of teaching and motivational methods.
- Supervised player development in the off-season - regulated practices and worked on specific skill sets with players.

Roland Lee, BSc, AIT

13020 SW Tapadera St. Beaverton, OR 97008
Online portfolio: http://server2.portland.iti.com/rlee

(503) 646-4467
rlee@graduates.iti.com

Technical project work:

The following projects were completed in an academic environment.

Web Application:

- Developed an interactive website for a Car broker using FrontPage 2000, VBScript, HTML, Javascript and Access 2000.
- Identified limits in capabilities of FrontPage 2000 and learned Macromedia Dreamweaver 4, Fireworks 3, and Flash 5 to enhance the appearance of the site Created a VBA application with Access 2000 of similar functionality as the website.

Visual Basic Application:

- Crowned "Debugging King" for ability to discern syntax and other unsolvable errors.
- Successful in integrating all of client's "extended features" to application.
- Built an application for a Placement Agency using Visual Basic 6.0 connected to an Access 2000 database using ADO and MTS.
- Developed a winning response to a Request For Proposal (RFP) that was both economically and strategically acceptable to the client.
- GUIDS Methodology and Unified Modeling Language (UML) techniques utilized and implemented to ensure good analysis and design for application completion. Designed and documented class diagram of 3-tier application using Microsoft Visual Modeler. Created a GUIDS document as reference for proper application development.
- Applied Object Oriented Programming in developing distributable components for client - separated application into executable component, business logic and data access dll's.
- Utilized Active X Components, SQL, and other Microsoft applications such as Outlook and Excel to further enhance the application's capabilities.
- Designed and integrated Active Server Pages to address client's desire to further distribute their application and make the website more interactive.

Java Application:

- Appointed by team to Project Technical lead.
- Developed a client/server application and applet for the Placement Agency in Java (Java and Java Beans) using JDK and IBM's Visual Age for Java interfaced with a legacy DB2 database using JDBC.
- Maintained strong Object-Oriented Analysis, Design and Application Development throughout process. Incorporated Model-View-Persistence design pattern in development of application.
- Utilized UML diagramming techniques such as CRC cards, Class Diagrams, and Sequence Diagrams to document and analyze application to further understand potential problems and solutions.
- Also implemented Extreme Programming techniques to facilitate efficient application development.

DB2:

- Designed, created, managed and deployed an enterprise scale RDBMS using DB2 UDB based on provided business narrative, functional specifications, and entity-attribute information.
- Utilized JDBC, SQL, JSP and Stored Procedures to create the back-end of a Java application as well as the web-based reports required by the client.
- Performed Database Administration responsibilities including: managing data, security, data integrity, backup/recovery, performance and maintenance.

Education:

Applied Information Technology **2001**
Post Graduate Diploma in AIT - November 2001
Information Technology Institute, Portland

Bachelor of Science **1997**
Biology Major/Chemistry Minor
Willamette University, Salem, OR

References

"He is an outstanding student and I highly recommend him both personally and professionally. As a student he is among a select few whom not only take the initiative to learn new and difficult concepts, but go on to master them completely. His understanding of the material was always thorough and in-depth. This understanding is only matched by his genuine personality.

Roland is very much the team player. He not only cares about the task at hand but takes care to make sure everyone on his team does as well. In the team environment Roland thrives on learning a subject and then teaching it back to his teammates. He is a creative, intelligent and thoughtful person who would be enormous asset to whomever he works with."

Ian I.D. Vanderberg
Former ITI Facilitator

Curriculum

iti™

Applied Information Technology AIT PROGRAM

"We use **collaborative** and **problem-based** learning methodologies to develop your technical, **problem-solving**, and **project management** skills, and prepare you to be a **valuable**, **team-oriented** member of the workplace. You'll develop your skills as you find solutions to real-world business problems as part of a project team, simulating an IT industry-based work environment."

The ITI curriculum is divided into five modules:

- ❑ Essentials of e-Business Computing
- ❑ Distributed Application Design and Development using Visual Basic 6
- ❑ Enterprise Application Development using Visual Age for Java
- ❑ Enterprise Application Development using DB2 UDB
- ❑ Professional Development

Several team projects are completed over the 9 month accelerated post-graduate program. Proper analysis and design are emphasized throughout the lifespan of each project. Also incorporated are team dynamic exercises (conflict resolution, team communication, stress management), delegation of tasks (sharing information, time management, application integration, resource and version management), and the production of technical documentation. Practical, team projects allow for applied learning of concepts learned during technical instruction. Presentations are made at the end of each project.

Team Project 1: Web Presence

"Students will design a Web Presence prototype for a business using **HTML**, FrontPage 2000, **VBScript** and other tools that they will research and use… Students will be required to develop forms and test the application developed for the Web. Students will be required to design and construct a relational database that allows users to publish and collect information over the Internet using **Access 2000**."

Team Project 2: Virtual Placement Agency:

"In the first phase of the project, students will build a **client/server database application** (1 or 2-tier) using the fundamental tools of the **Visual Basic 6.0** Integrated Development Environment (IDE).

Students will then take the completed simple client/server project and further develop the project in an n-tier environment. The concepts of **Object-Oriented Analysis and Design** will be introduced, and students will apply them in the enhancement of their client/server application in an **n-tier** environment… Microsoft Visual Modeler will be used in the creation of a complete set of GUIDS documents.

Once students have designed the project enhancements, they will produce Visual Basic 6.0 source code using class modules. Students will be introduced to and use **ADO**, **SQL**, **ActiveX**, and **MTS** technologies to develop the enhanced project.

In the final phase of the project, students will deploy an enhanced n-tier application across a LAN and the web using **ASP** and **IIS** technologies. "

Team Project 3: Personnel Placement Agency

"As a model for a large scale, real-life application, this project allows students to demonstrate proficiency in Java Technology using Visual Age. Object-oriented design and development will help students gain an understanding of the potential of the Internet for business applications. Students will use a Virtual Placement Agency problem from Visual Basic and create a Java-based solution. Students will discover the potential for powerful 3-tier architecture enterprise computing capabilities."

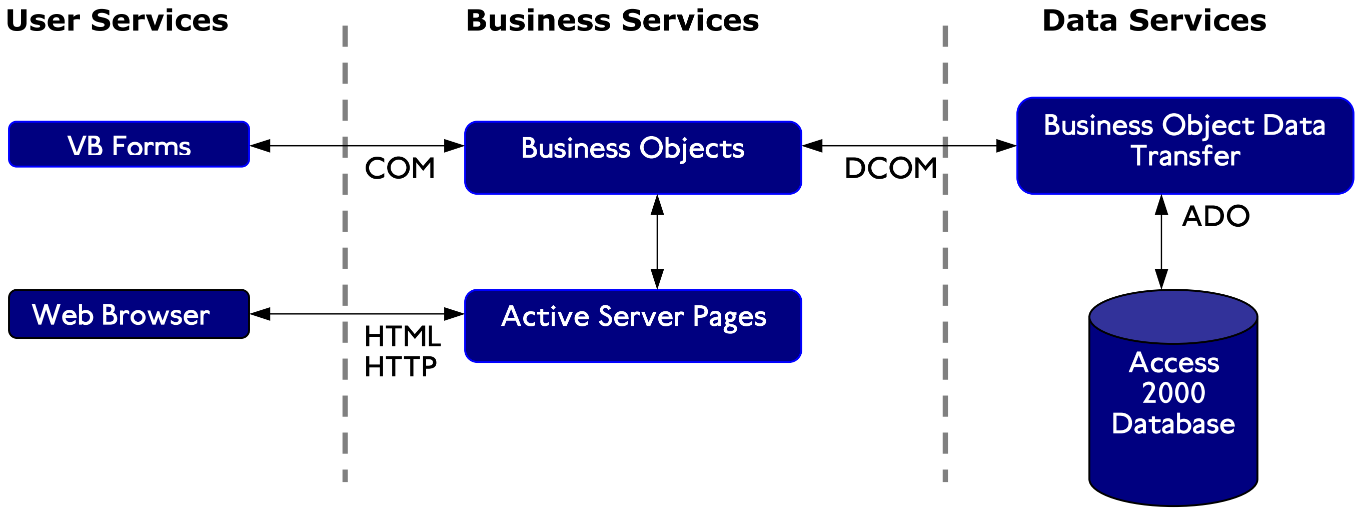
Team Project 4: 'Imagine' Project Tracking System

Phase 1: " examine enterprise development, the various DB2 UDB tools used throughout the module, creation of DB2 users through NT User Manager, database design, SQL, and database object creation."
Phase 2: " apply acquired java skills to use JDBC, Java Server Pages and Stored Procedures to accomplish the database connectivity."
Phase 3: "This phase will expose you to various database administration tasks such as security, managing your data, data integrity, backup and recovery, performance and maintenance."

Visual Basic 6

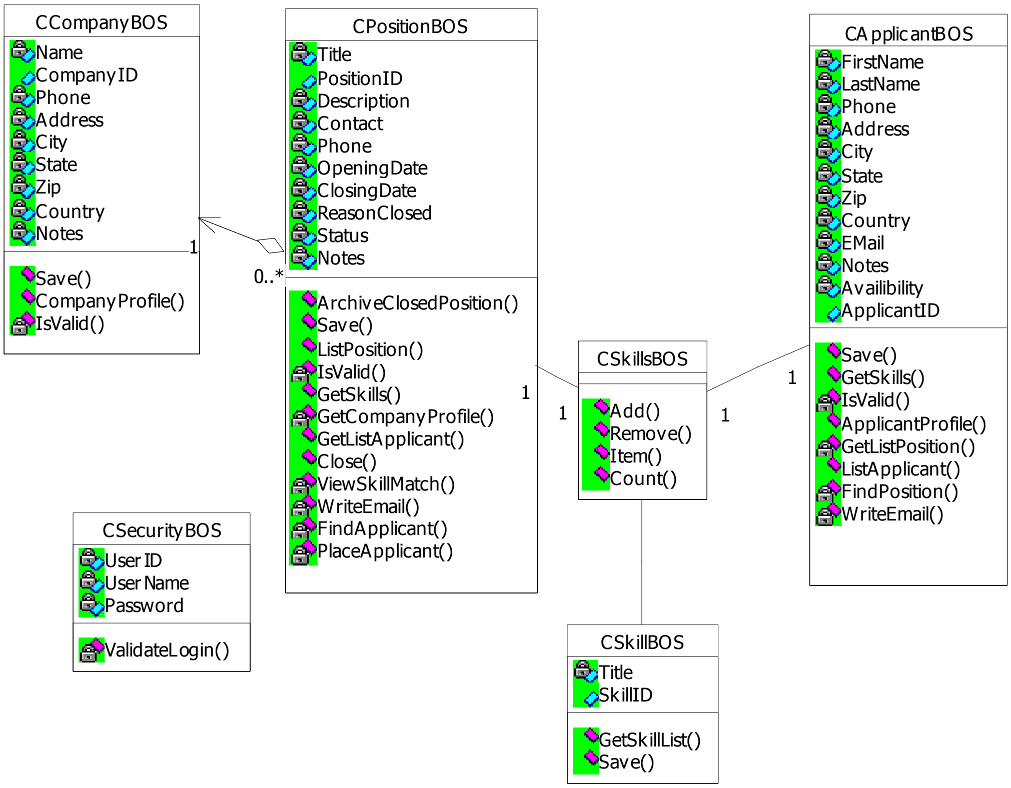
Project Description

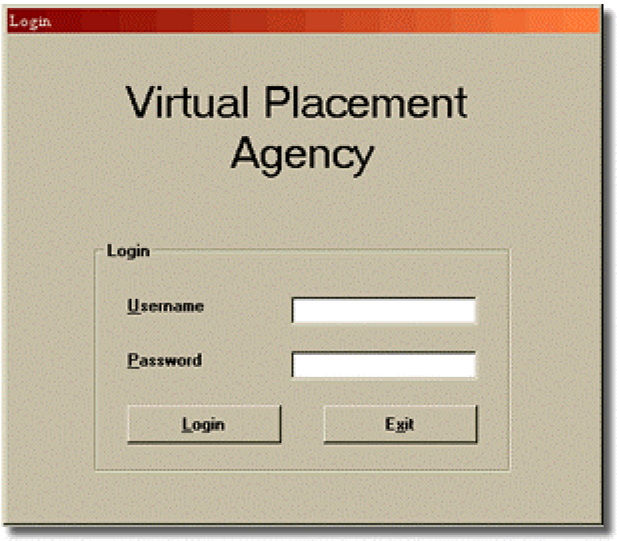
❑ Developing a distributable client/server application using Microsoft Visual Basic 6.0. Deployed across the Internet and connected to a Microsoft Access 2000 back end using COM+, MTS, and IIS.
❑ Diagramming models using UML.
❑ Using GUIDS methodology to create documentation for use in application development.
❑ Developing GUI using proper GUI standards.
❑ Using object-oriented analysis and design as well as component-based development.
❑ Creating Active Server Pages and using VBScript.



The **Logical Design Pattern** (Above) is not concerned with physical implementation, but rather defining the abstractions necessary to achieve the appropriate scenarios with the components of the application

The **Class Diagram** (Right) is a visual tool that assists in documenting the objects including their properties and methods and to define the relationships between the objects including associations, aggregations, and multiplicity.





Login

Virtual Placement Agency

Login

Username

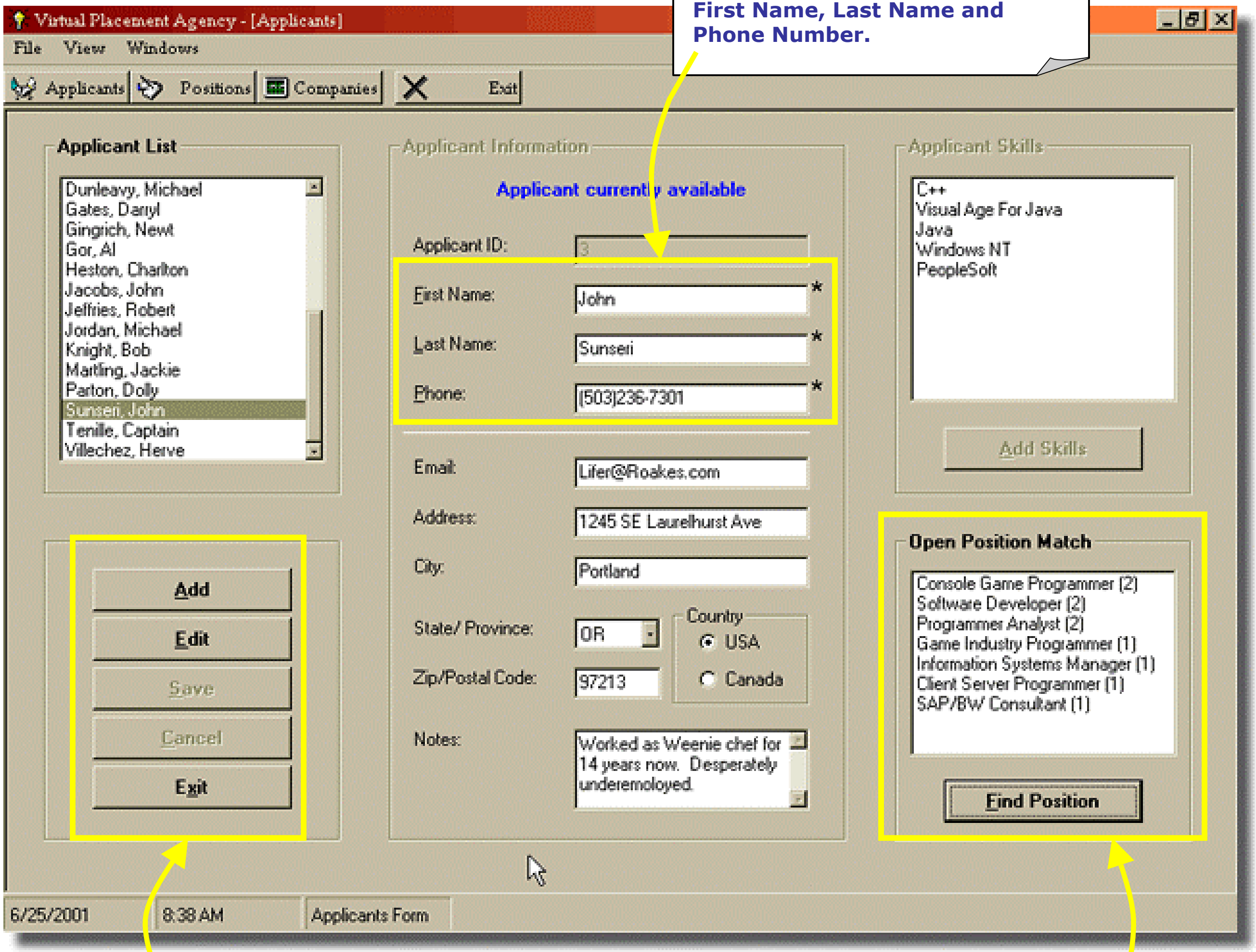
Password

Login Exit

(Left) Upon starting the application, the user is required to login using a username and password. Upon successful login, the main MDI form functions are enabled and user can proceed.

(Below) Here is a sample of the Applicants form. The user can add, edit, applicants to the database.

Our business rules state that an applicant should at least have a First Name, Last Name and Phone Number.



Virtual Placement Agency - [Applicants]

File View Windows

Applicants Positions Companies Exit

Applicant List

- Dunleavy, Michael
- Gates, Darryl
- Gingrich, Newt
- Gor, Al
- Heston, Charlton
- Jacobs, John
- Jeffries, Robert
- Jordan, Michael
- Knight, Bob
- Martling, Jackie
- Parton, Dolly
- Sunseri, John
- Tenille, Captain
- Villechez, Herve

Add
Edit
Save
Cancel
Exit

Applicant Information

Applicant currently available

Applicant ID: 3

First Name: John *

Last Name: Sunseri *

Phone: (503)236-7301 *

Email: Lifer@Roakes.com

Address: 1245 SE Laurelhurst Ave

City: Portland

State/ Province: OR

Country: ● USA ○ Canada

Zip/Postal Code: 97213

Notes: Worked as Weenie chef for 14 years now. Desperately underemoloyed.

Applicant Skills

- C++
- Visual Age For Java
- Java
- Windows NT
- PeopleSoft

Add Skills

Open Position Match

- Console Game Programmer (2)
- Software Developer (2)
- Programmer Analyst (2)
- Game Industry Programmer (1)
- Information Systems Manager (1)
- Client Server Programmer (1)
- SAP/BW Consultant (1)

Find Position

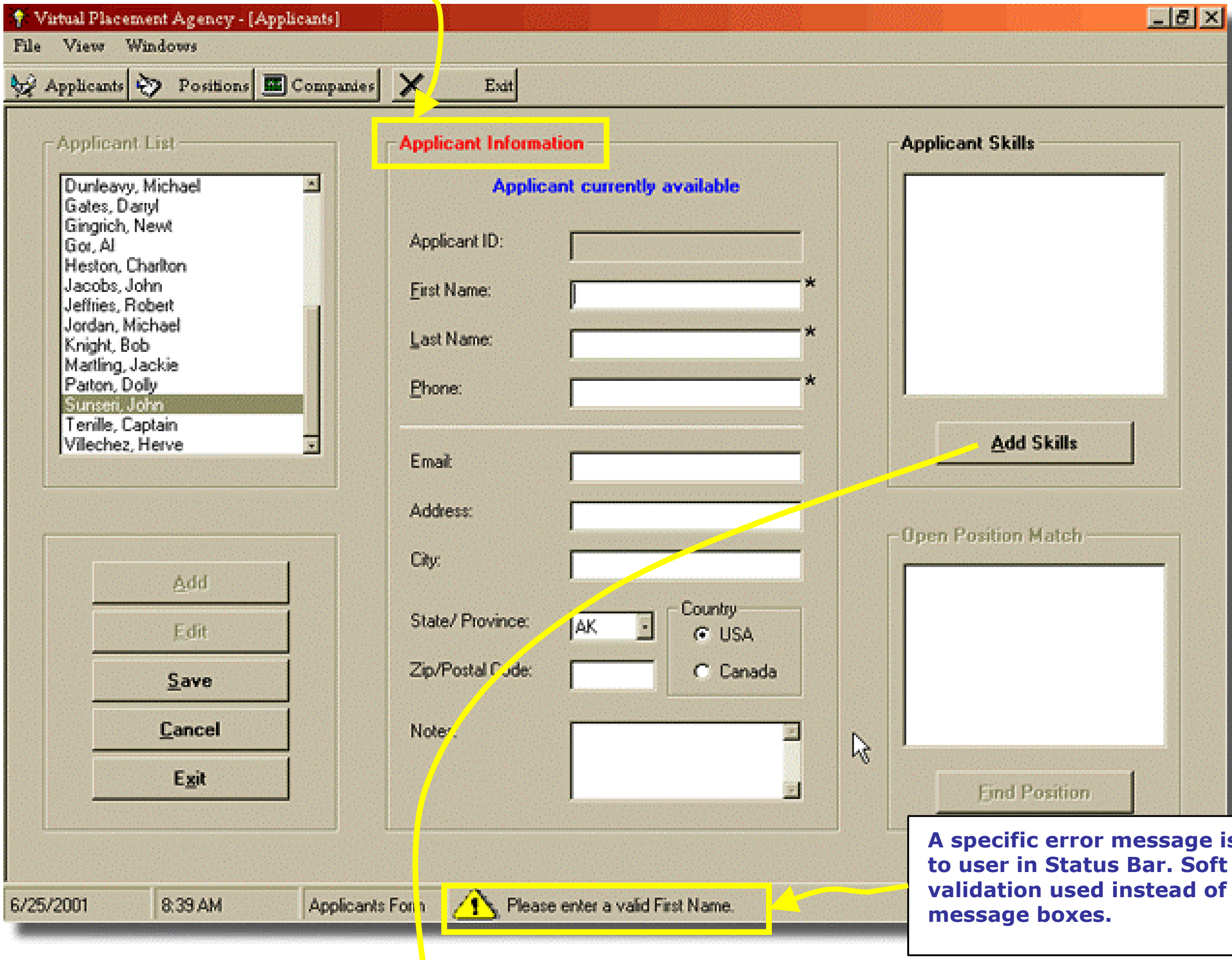
6/25/2001 8:38 AM Applicants Form

Ease-of-use and functionality of application was emphasized throughout application. Here, when viewing applicants, only specific buttons are available.

For available Applicants the user is able to find open positions, which are sorted by number of skills the applicant matches for that specific position.

The frame where the error originated is highlighted in red. Focus is also set on specific error.

Upon clicking add, the applicant's status is set to available and all form elements are emptied. Here is a sample situation of error handling/form validation that was implemented.



Dunleavy, Michael
Gates, Daryl
Gingrich, Newt
Gor, Al
Heston, Charlton
Jacobs, John
Jeffries, Robert
Jordan, Michael
Knight, Bob
Marling, Jackie
Parton, Dolly
Sunseri, John
Terrille, Captain
Villechez, Herve

Applicant currently available

Applicant ID:

First Name: *

Last Name: *

Phone: *

Email:

Address:

City:

State/ Province: AK

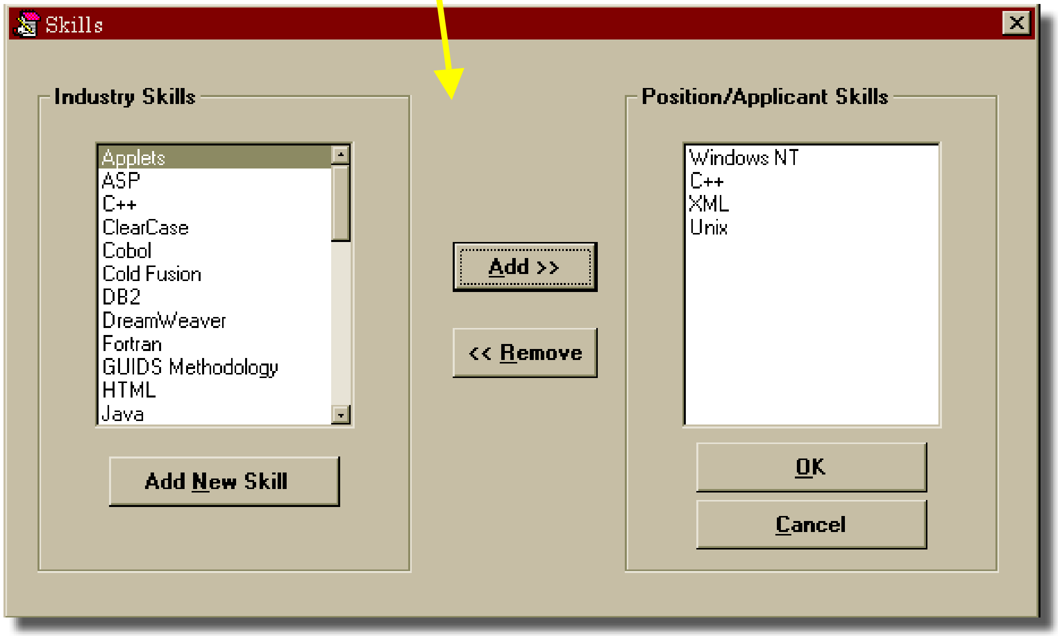
Country: ● USA ○ Canada

Zip/Postal Code:

Notes:

6/25/2001 | 8:39 AM | Applicants Form | Please enter a valid First Name.

A specific error message is shown to user in Status Bar. Soft validation used instead of message boxes.

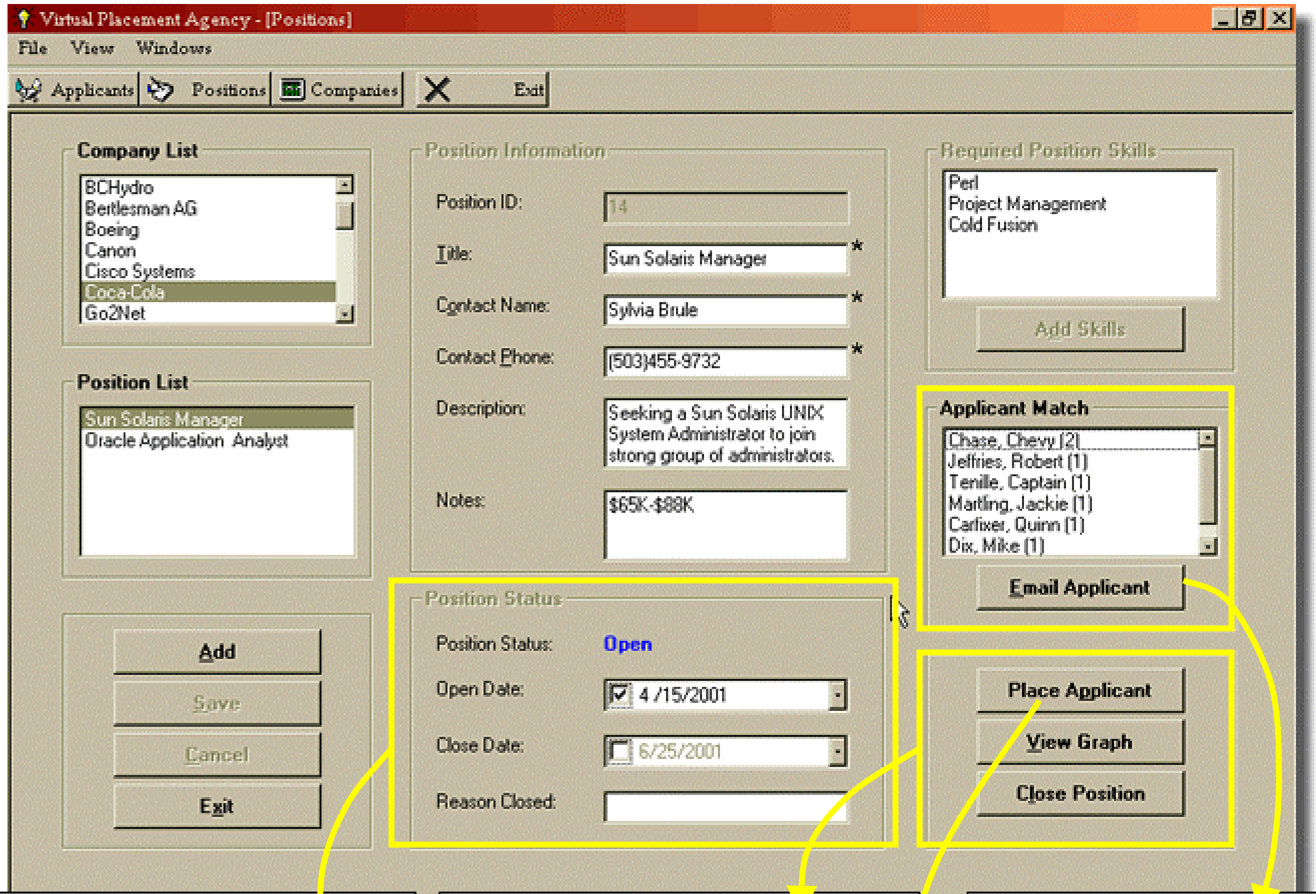


Industry Skills:
Applets
ASP
C++
ClearCase
Cobol
Cold Fusion
DB2
DreamWeaver
Fortran
GUIDS Methodology
HTML
Java

Position/Applicant Skills:
Windows NT
C++
XML
Unix

Upon clicking add skills, the user can add, remove industry skills to the applicant's skill set.

If skill is not found in list, user has the option to add a new skill.



Company List

BCHydro
Bertlesman AG
Boeing
Canon
Cisco Systems
Coca-Cola
Go2Net

Position List

Sun Solaris Manager
Oracle Application Analyst

Position Information

Position ID: 14

Title: Sun Solaris Manager *

Contact Name: Sylvia Brule *

Contact Phone: (503)455-9732 *

Description: Seeking a Sun Solaris UNIX System Administrator to join strong group of administrators.

Notes: $65K-$88K

Position Status

Position Status: Open

Open Date: ☑ 4 /15/2001

Close Date: □ 6/25/2001

Reason Closed:

Required Position Skills

Perl
Project Management
Cold Fusion

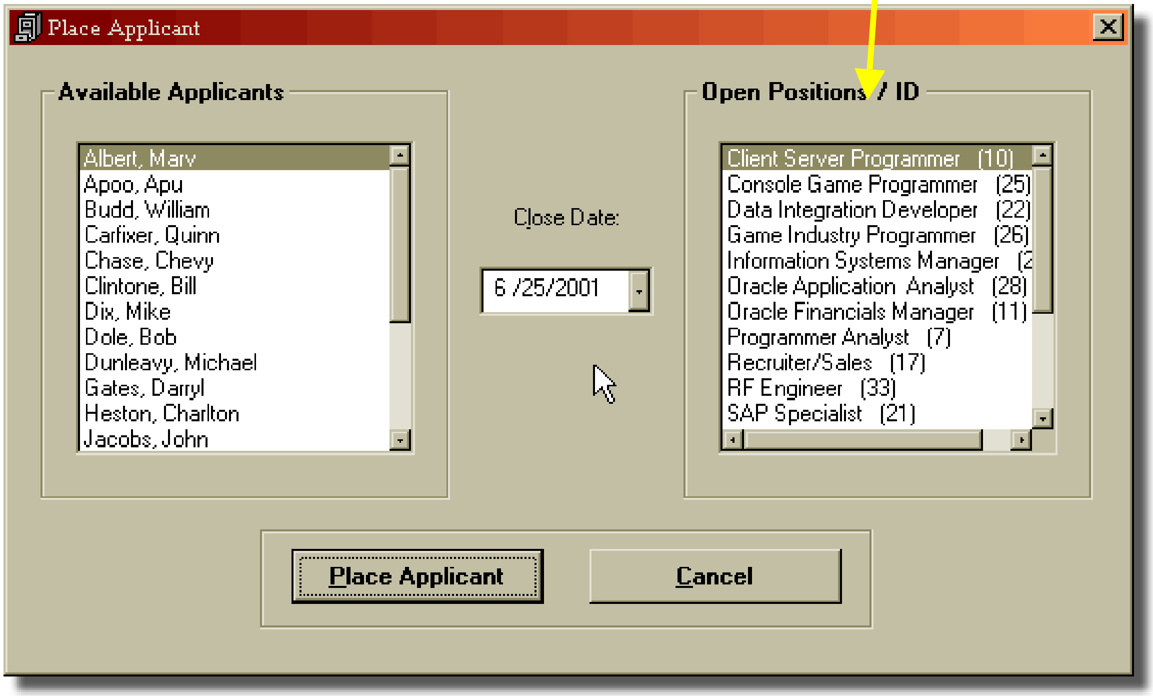
Applicant Match

Chase, Chevy (2)
Jelfries, Robert (1)
Tenille, Captain (1)
Marling, Jackie (1)
Carfixer, Quinn (1)
Dix, Mike (1)

Each position has a position status. To close a position, users must supply a close date and a reason for closed. A component called a DateTimePicker was used for date validation.

Other options the user has include placing applicants who are available with open positions. An Excel Pie Chart showing the percent of applicants matching the skills of the position can be selected. User can also close an open position.

Users can view applicants that match the skill set of the specific position. User can email applicants with email addresses. Outlook opens in the background and sends messages for the user.



Available Applicants

Albert, Marv
Apoo, Apu
Budd, William
Carfixer, Quinn
Chase, Chevy
Clintone, Bill
Dix, Mike
Dole, Bob
Dunleavy, Michael
Gates, Darryl
Heston, Charlton
Jacobs, John

Close Date: 6 /25/2001

Open Positions / ID

Client Server Programmer (10)
Console Game Programmer (25)
Data Integration Developer (22)
Game Industry Programmer (26)
Information Systems Manager (2
Oracle Application Analyst (28)
Oracle Financials Manager (11)
Programmer Analyst (7)
Recruiter/Sales (17)
RF Engineer (33)
SAP Specialist (21)

(Above) Users can add new positions to the database for existing companies.

(Left)
When placing an applicant, a list of available applicants and open positions is displayed. Users match an applicant and a position and select a closing date.

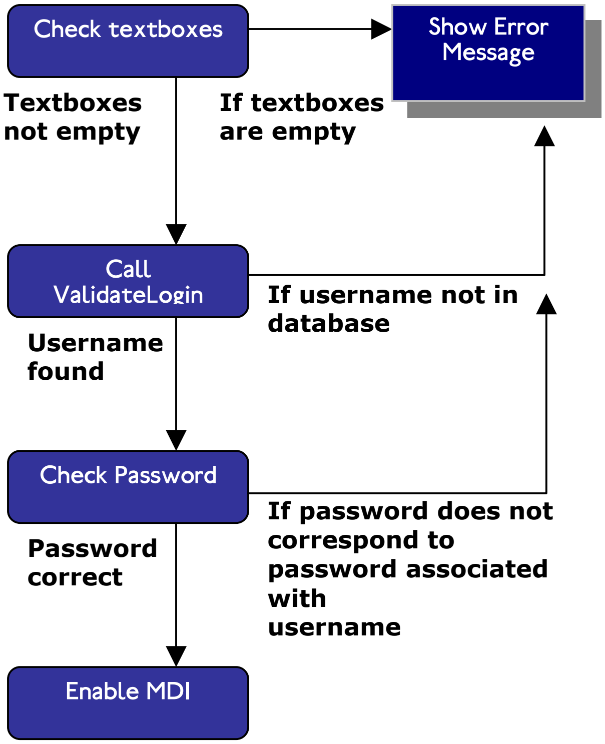
This method e-mails applicants matched to a specific position. Only applicants with email addresses are sent emails and after the task is done, a message on the status bar is shown indicating how many emails were sent.

```
Private Sub cmdEmail_Click()
  'after applicants are matched with a position, option to e-mail
  'each applicant regarding position.
  'at end of process, message indicates number of emails sent

On Error GoTo errorHandler
  Dim objOutlook As New Outlook.Application
  Dim objOutlookMsg As Outlook.MailItem
  Dim arrApplicants As Variant
  Dim intCounter As Integer
  Dim strPosition As String
  Dim strEmail As String
  Dim strFirstName As String
  Dim strLastName As String
  Dim strCompany As String
  Dim intMessageSent As Integer
  Dim intMessageNotSent As Integer
  'set array receiving info from BODT
arrApplicants=mobjPosition.MatchApplicantEmail(Me.lstOpenPosition.ItemData(Me.lstOpenPosition.ListIndex))
  'use for loop to go through array and hold info in variables
   For intCounter = 0 To UBound(arrApplicants, 2)
    strFirstName = arrApplicants(2, intCounter)
    strLastName = arrApplicants(1, intCounter)
    strEmail = arrApplicants(3, intCounter)
  'fill variables with position, company info from private variable, list box
    strPosition = mobjPosition.Title
    strCompany = Me.lstCompany.List(Me.lstCompany.ListIndex)
  'Check if applicant has email address in applicant information
  'If so, send an email message
    If strEmail <> "" Then
  'Instantiate objects
      Set objOutlookMsg = objOutlook.CreateItem(olMailItem)
  'Write message
      With objOutlookMsg
       .To = strEmail
       .Subject = "Position Match" & ": " & strPosition
       .Body = "Dear " & strFirstName & " " & strLastName & "," & vbNewLine & vbNewLine
       .Body = .Body & "I have found a position that matches one or more of your job skills: " &
       vbNewLine & vbNewLine
       .Body = .Body & "Position Title: " & strPosition & vbNewLine
       .Body = .Body & "Company: " & strCompany & vbNewLine & vbNewLine
       .Body = .Body & "I will be contacting you shortly regarding the position. "
       .Body = .Body & "At that time I can provide you with more details about "
       .Body = .Body & "the position, and help you determine if it is one "
       .Body = .Body & "you wish to pursue." & vbNewLine & vbNewLine
       .Body = .Body & "Sincerely," & vbNewLine & vbNewLine
       .Body = .Body & "Alan Gregory" & vbNewLine
       .Body = .Body & "Placement Associate" & vbNewLine
       .Body = .Body & "Virtual Placement Agency" & vbNewLine
       .Body = .Body & "agregory@vpa.com" & vbNewLine
       .Body = .Body & "(800) 555-0987"
       .Send
      End With
  'track number of emails sent to applicant
      intMessageSent = intMessageSent + 1
    Else
  'track number of applicants without email
      intMessageNotSent = intMessageNotSent + 1
    End If
  Next
  'display message indicating number of messages sent,
  'and how many applicants did not have an email address to send to
  frmMDI.stamdi.Panels(4).Visible = True
  frmMDI.stamdi.Panels(4).Picture = frmMDI.ilsForms.ListImages(2).Picture
  frmMDI.stamdi.Panels(4).Text = intMessageSent & " message(s) were sent to applicants. " &
  intMessageNotSent & " applicant(s) did not have an email address."
  'destroy objects
   Set objOutlookMsg = Nothing
   Set objOutlook = Nothing
  'reset visibility of layered buttons
   cmdMatchApplicant.Visible = True
   cmdemail.Visible = False
Exit Sub
errorHandler:
Call CheckError

End Sub
```

```
Private Sub cmdLogin_Click()
'this sub on click checks the user name and password text boxes for values.
'If values are present then connect to database and find username and password.

Dim arrSecurity As Variant
On Error GoTo errorHandler
'check textboxes for values
  If Me.txtUsername.Text = "" Then
    frmMDI.stamdi.Panels(4).Visible = True
    frmMDI.stamdi.Panels(4).Picture = frmMDI.ilsForms.ListImages(1).Picture
    frmMDI.stamdi.Panels(4).Text = "Please enter a Username."
    Me.txtUsername.SetFocus
  ElseIf Me.txtPassword.Text = "" Then
    frmMDI.stamdi.Panels(4).Visible = True
    frmMDI.stamdi.Panels(4).Picture = frmMDI.ilsForms.ListImages(1).Picture
    frmMDI.stamdi.Panels(4).Text = "Please enter a Password."
    Me.txtPassword.SetFocus
  Else
'call validateLogin method in Security Module, passing username
'if user name is not present, error will be raised by module
    Call mobjSecurity.ValidateLogin(Me.txtUsername.Text)
'check password variable value and compare with password textbox
'if password is incorrect, show message in status bar
'else enable MDI form
   If Me.txtPassword.Text = mobjSecurity.Password Then
    frmMDI.mnufile.Enabled = True
    frmMDI.mnuview.Enabled = True
    frmMDI.mnuwindows.Enabled = True
    frmMDI.tlbmdi.Enabled = True
    Unload frmLogin
    Call ClearStatusbar
   Else
    Err.Raise LoginErrors.InvalidPassword, "Login", "Invalid Password"
   End If
  End If
Exit Sub

errorHandler:

Call SecurityErrors

End Sub
```

(Left)
This method was written for the login screen.

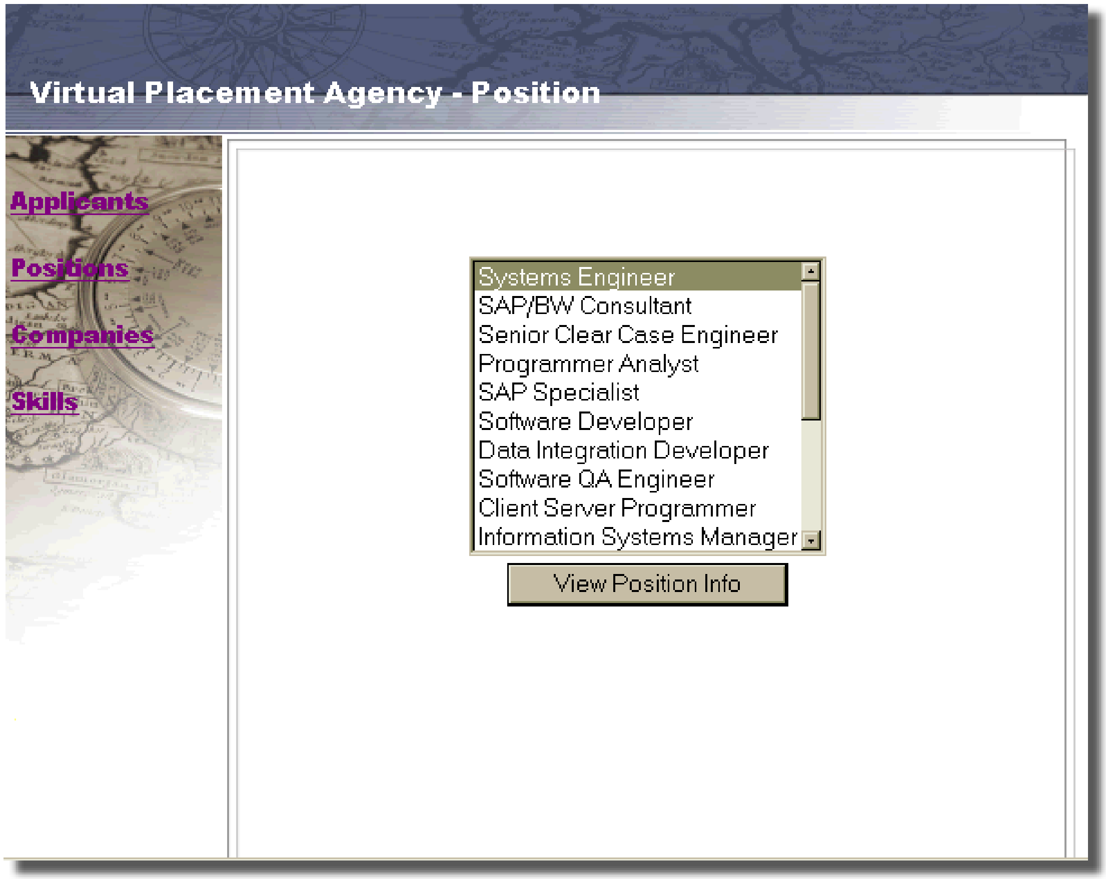
(Below)
A specific logical sequence was required as diagrammed below.



(Below)
This method queries the database for matching positions for an applicant's skill set.
Positions are ordered by number of skill matches for the positon.

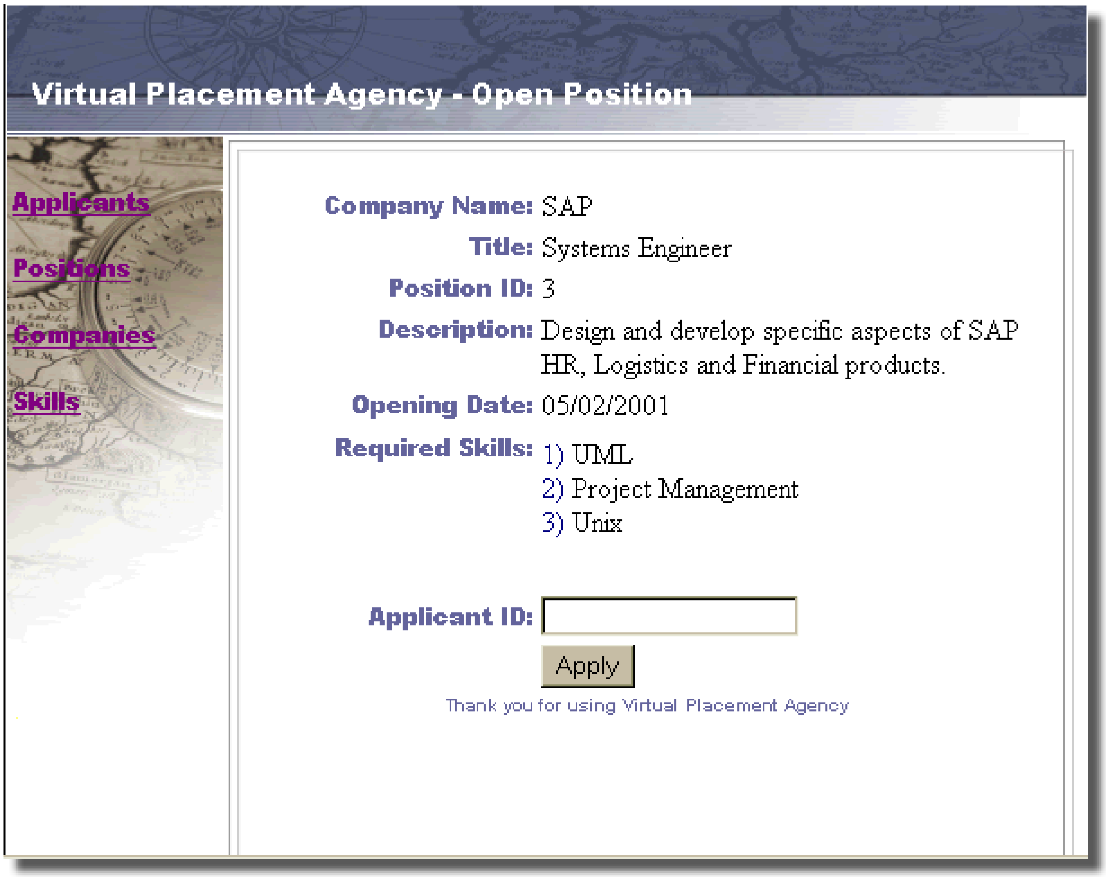
```
Public Function FindPosition(intId As Integer) As Variant
'This function returns a variant array with matching positions for an applicant

  Dim strSQL As String
  'select position IDs and titles from tblposition and
   strSQL = "SELECT tblPositionSkill.PositionID, tblPosition.Title, Count(tblPositionSkill.SkillID) AS CountOfSkillID "

  'inner join applicantskill and positionskill tables where applicant's skill id = positionskill's skill ids
   strSQL = strSQL & "FROM tblPosition INNER JOIN (tblApplicantSkill "
   strSQL = strSQL & " INNER JOIN tblPositionSkill ON tblApplicantSkill.SkillID = tblPositionSkill.SkillID) "

  'inner join position and positionskill table data where positionid's match
   strSQL = strSQL & "ON tblPosition.PositionID = tblPositionSkill.PositionID "

  'group by positionID, title, status, and applicantID
   strSQL = strSQL & "GROUP BY tblPositionSkill.PositionID, tblPosition.Title, tblPosition.Status, tblApplicantSkill.ApplicantID "

  'having status open and applicant id is passed id
   strSQL = strSQL & "HAVING (((tblPosition.Status)='open') AND ((tblApplicantSkill.ApplicantID)=" & intId & "))"

  'order by count of skillids- greater first or best fit
   strSQL = strSQL & "ORDER BY Count(tblPositionSkill.SkillID) DESC"

   mobjdataaccess.Connect
   FindPosition = mobjdataaccess.RetrieveData(strSQL)
   mobjdataaccess.Disconnect

End Function
```

Active Server Pages



(Left) Registered applicants can Login via the web and view open positions.

(Below) Selecting a position and clicking the 'View Position Info' displays the position's information. If the applicant is interested in the position they can enter their ID and select the 'Apply' option.



Company Name: SAP

Title: Systems Engineer

Position ID: 3

Description: Design and develop specific aspects of SAP HR, Logistics and Financial products.

Opening Date: 05/02/2001

Required Skills:
1) UML
2) Project Management
3) Unix

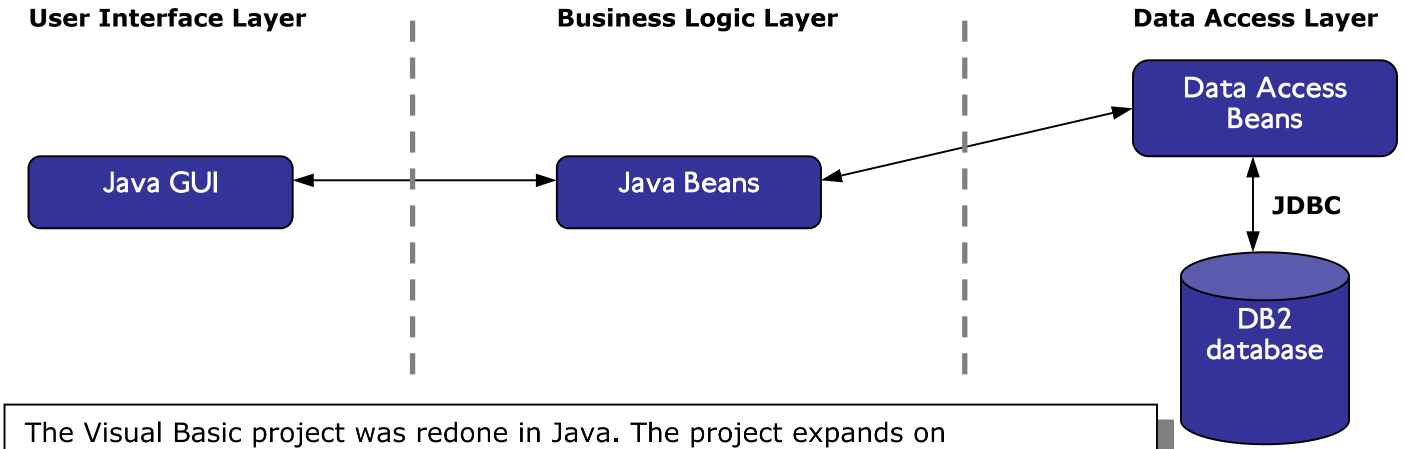
Applicant ID:

Thank you for using Virtual Placement Agency

Java

Project Description

- Developing a client/server application using Visual Age for Java IDE and JDK.
- Developing a response to a Request-For-Proposal (RFP).
- Utilizing UML diagramming techniques, Object-Oriented design and analysis, Object-oriented programming to design and develop application.
- Creating reusable components called Java Beans.
- Using Object-Oriented design patterns in creating application.
- Deploying application and creating an applet.
- Writing a user manual and generating documentation through JavaDoc.



User Interface Layer

Business Logic Layer

Data Access Layer

Java GUI

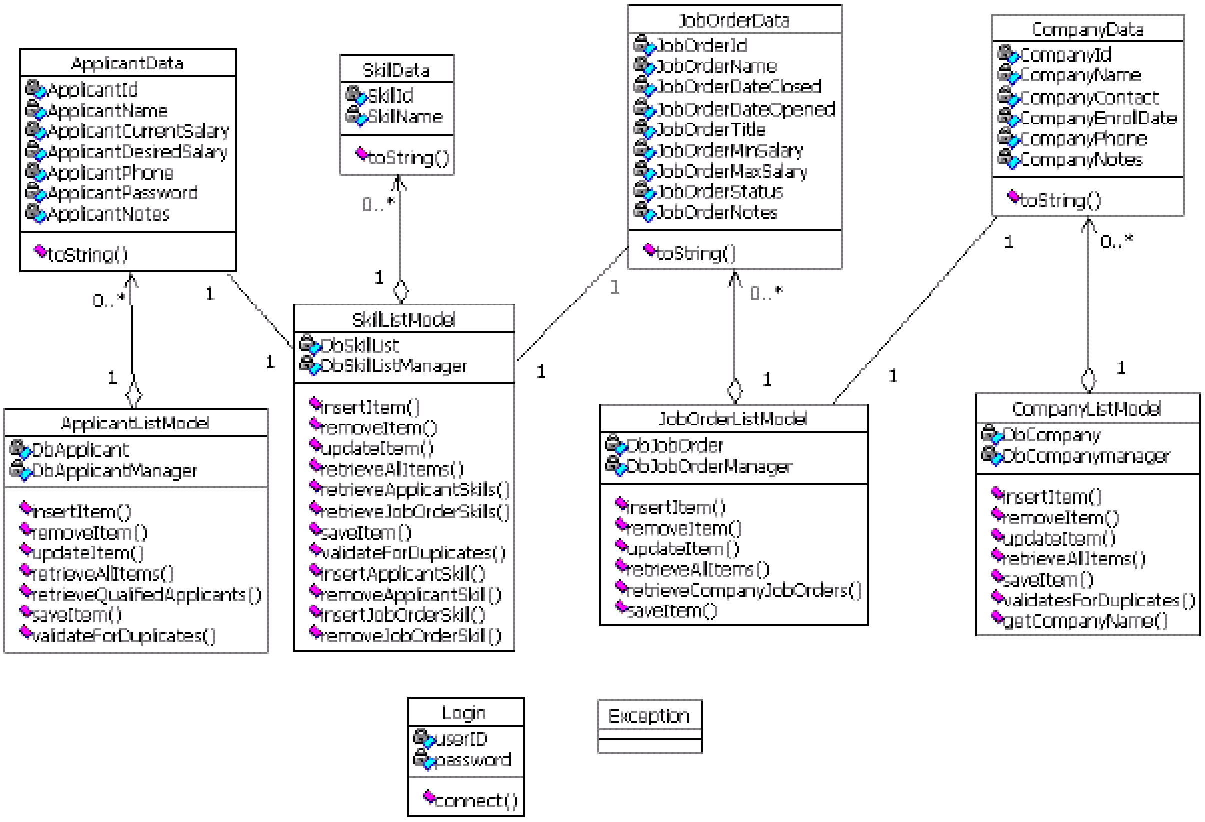
Java Beans

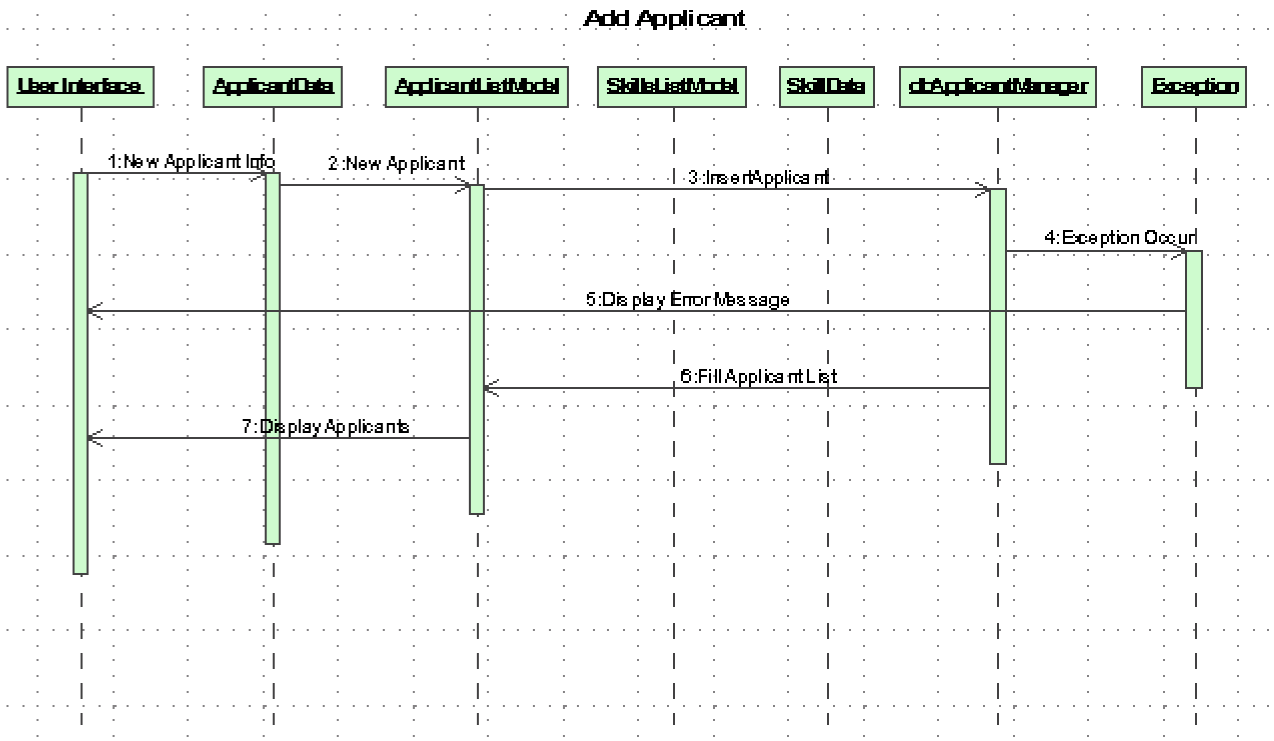
Data Access Beans

JDBC

DB2 database

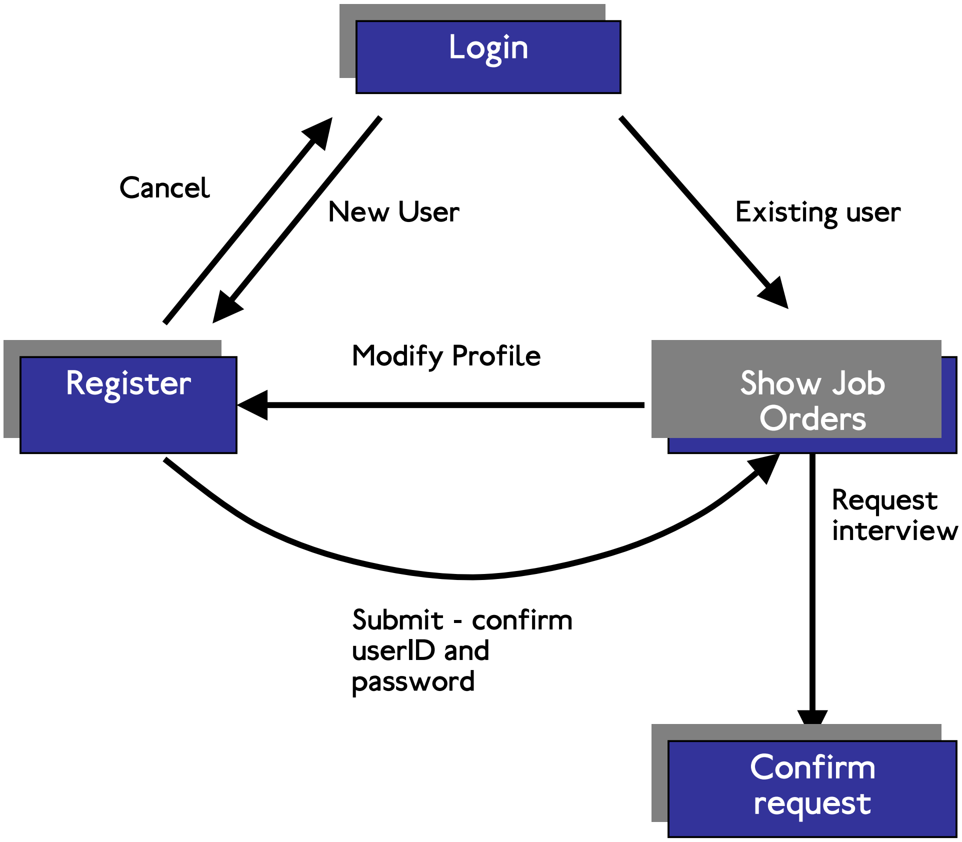
The Visual Basic project was redone in Java. The project expands on Object-Oriented Analysis & Design by utilizing Unified Modeling Language (UML) diagramming techniques. (Above) A graphical representation of the n-tier application.
(Below) Object diagram created based on abstractions of business narrative.

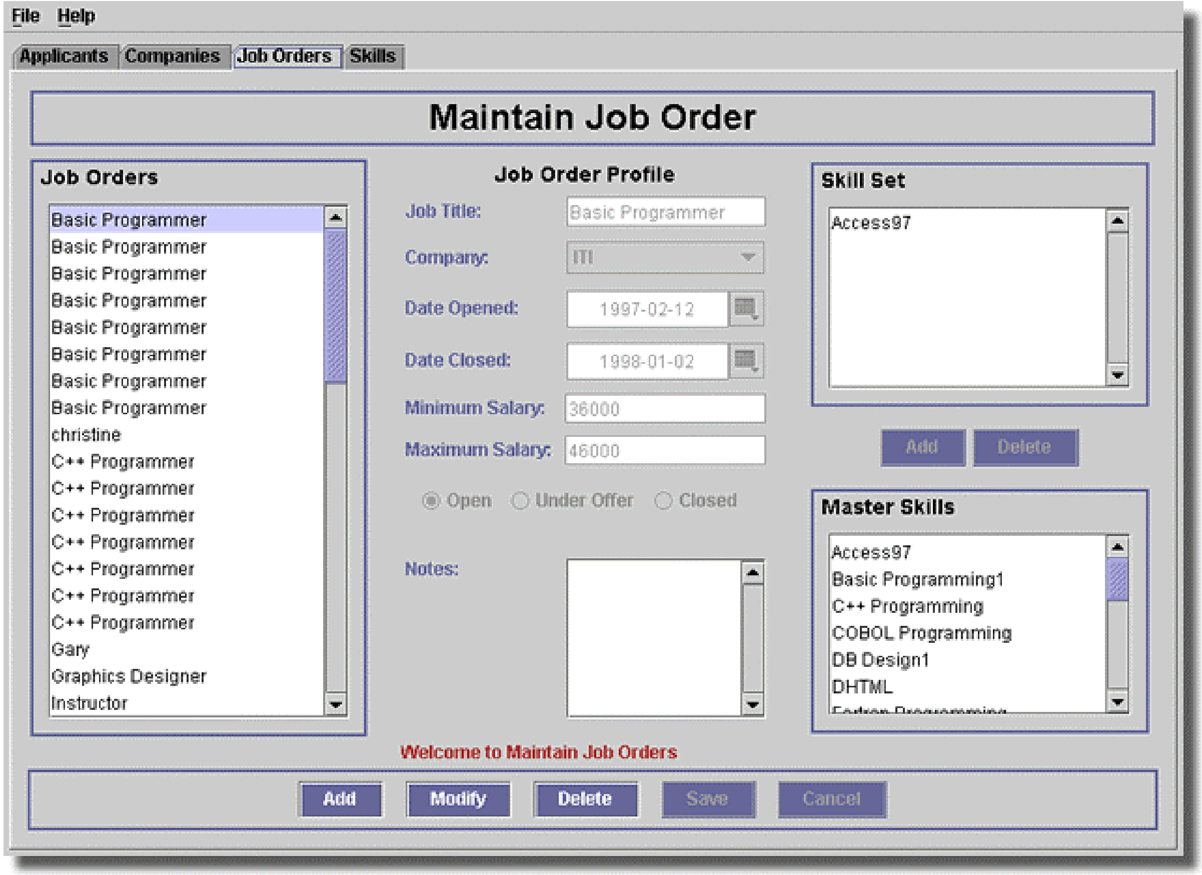




(Above)
The sequence diagram is a visual interpretation of how a specific Use Case is implemented. This includes indicating responsibilities of each class and the communication between them.

(Right)
The applet storyboard details the different ways the user can interact on the web with the client's system.





Maintain Job Order

Job Orders

Basic Programmer
Basic Programmer
Basic Programmer
Basic Programmer
Basic Programmer
Basic Programmer
Basic Programmer
Basic Programmer
christine
C++ Programmer
C++ Programmer
C++ Programmer
C++ Programmer
C++ Programmer
C++ Programmer
C++ Programmer
Gary
Graphics Designer
Instructor

Job Order Profile

Job Title: Basic Programmer
Company: ITI
Date Opened: 1997-02-12
Date Closed: 1998-01-02
Minimum Salary: 36000
Maximum Salary: 46000
● Open ○ Under Offer ○ Closed
Notes:

Skill Set

Access97

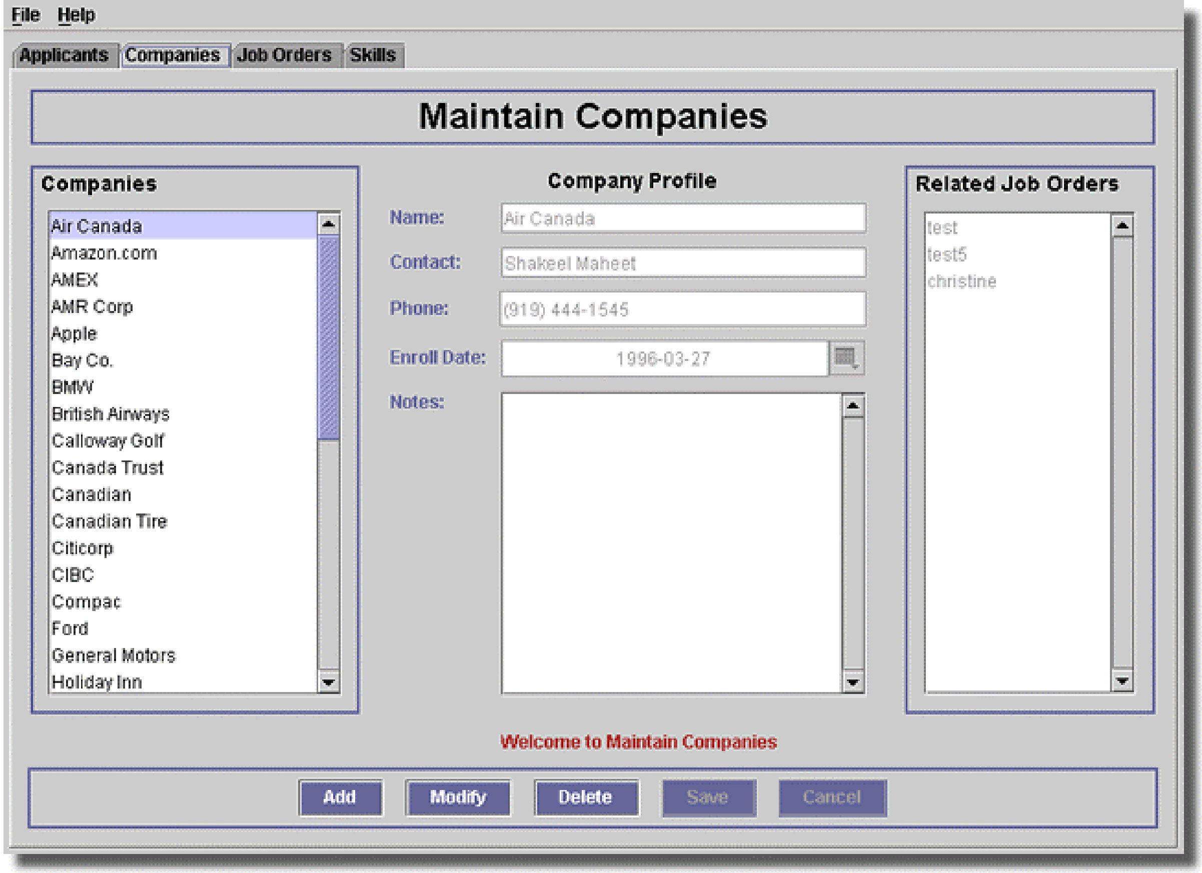
Master Skills

Access97
Basic Programming1
C++ Programming
COBOL Programming
DB Design1
DHTML
Fortran Programming

Welcome to Maintain Job Orders

The user can view a list of Job Orders. They also have the option of adding, editing, deleting a job order records.

For ease of use, a combo box with all the companies is used during entry of a job record. Other things include radio buttons for the status of the job, text fields that only allow numbers for the salary, and a drop-down calendar to select dates.



Maintain Companies

Companies

Air Canada
Amazon.com
AMEX
AMR Corp
Apple
Bay Co.
BMW
British Airways
Calloway Golf
Canada Trust
Canadian
Canadian Tire
Citicorp
CIBC
Compac
Ford
General Motors
Holiday Inn

Company Profile

Name: Air Canada
Contact: Shakeel Maheet
Phone: (919) 444-1545
Enroll Date: 1996-03-27
Notes:

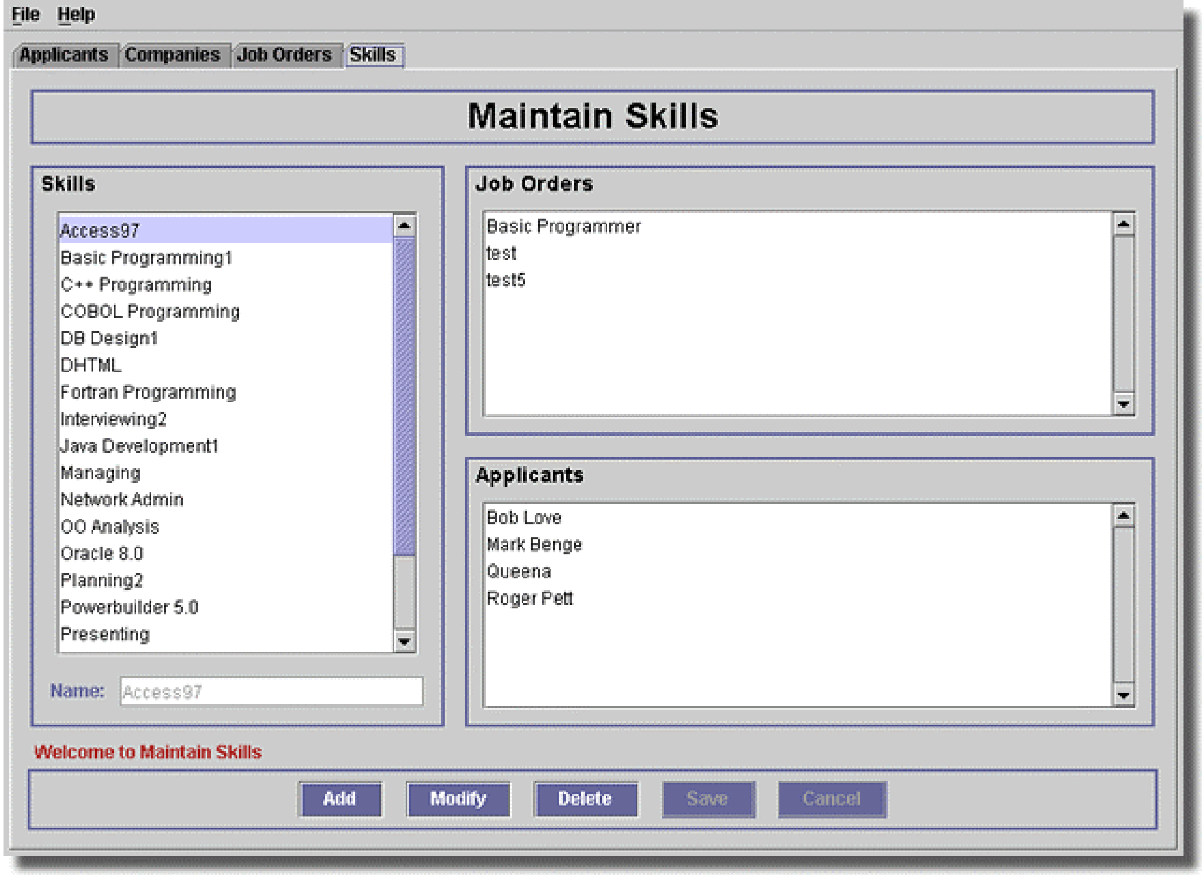
Related Job Orders

test
test5
christine

Welcome to Maintain Companies

Here the user can perform similar functionality on companies in their database. A list of related job orders are listed for the selected company. Companies with job orders cannot be deleted.

A text field specific for phone numbers is used to ensure proper data is entered.



Maintain Skills

Skills

Access97
Basic Programming1
C++ Programming
COBOL Programming
DB Design1
DHTML
Fortran Programming
Interviewing2
Java Development1
Managing
Network Admin
OO Analysis
Oracle 8.0
Planning2
Powerbuilder 5.0
Presenting

Name: Access97

Job Orders

Basic Programmer
test
test5

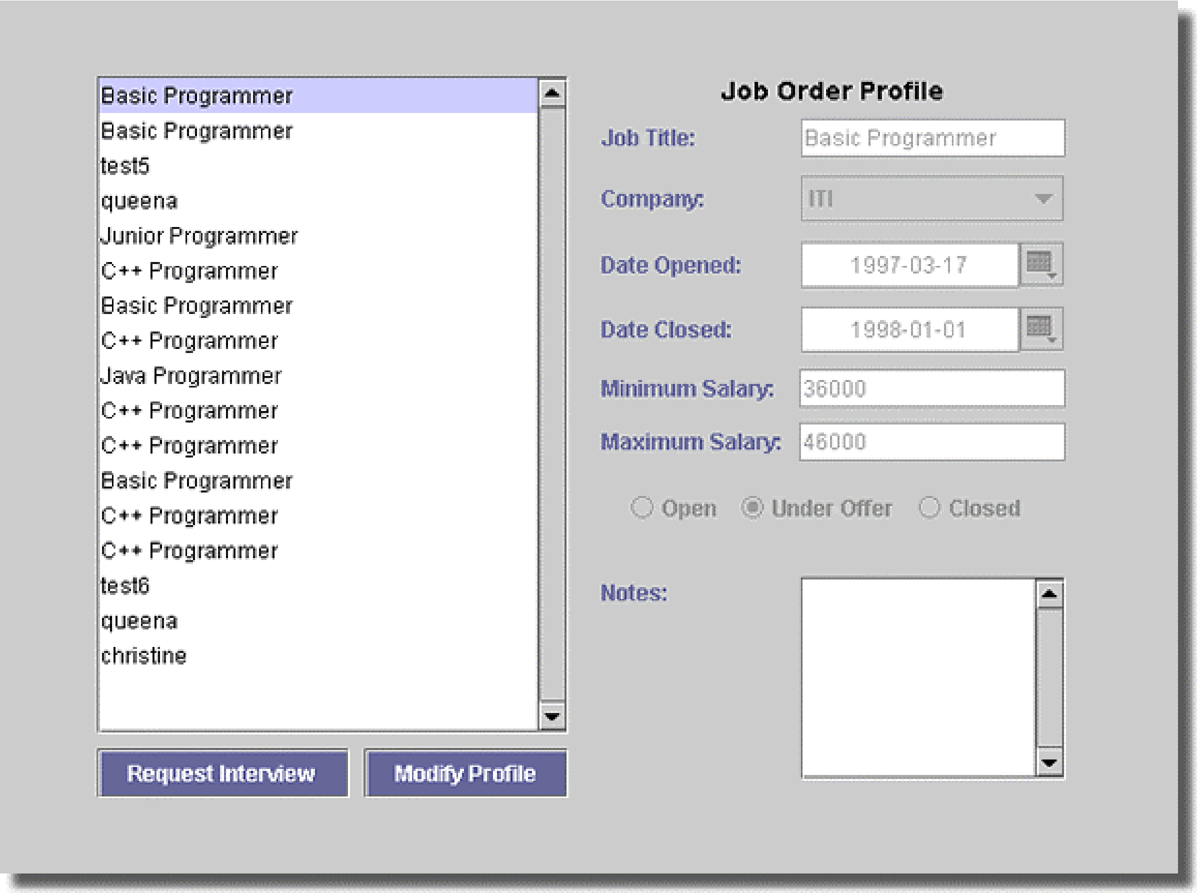
Applicants

Bob Love
Mark Benge
Queena
Roger Pett

Welcome to Maintain Skills

This screen shows the master list of skills in the database.

Users can add, edit, and delete skills, unless a user has the specific skill. Also listed are the applicants and job orders that have the selected skill.



Basic Programmer
Basic Programmer
test5
queena
Junior Programmer
C++ Programmer
Basic Programmer
C++ Programmer
Java Programmer
C++ Programmer
C++ Programmer
Basic Programmer
C++ Programmer
C++ Programmer
test6
queena
christine

Job Order Profile

Job Title: Basic Programmer

Company: ITI

Date Opened: 1997-03-17

Date Closed: 1998-01-01

Minimum Salary: 36000

Maximum Salary: 46000

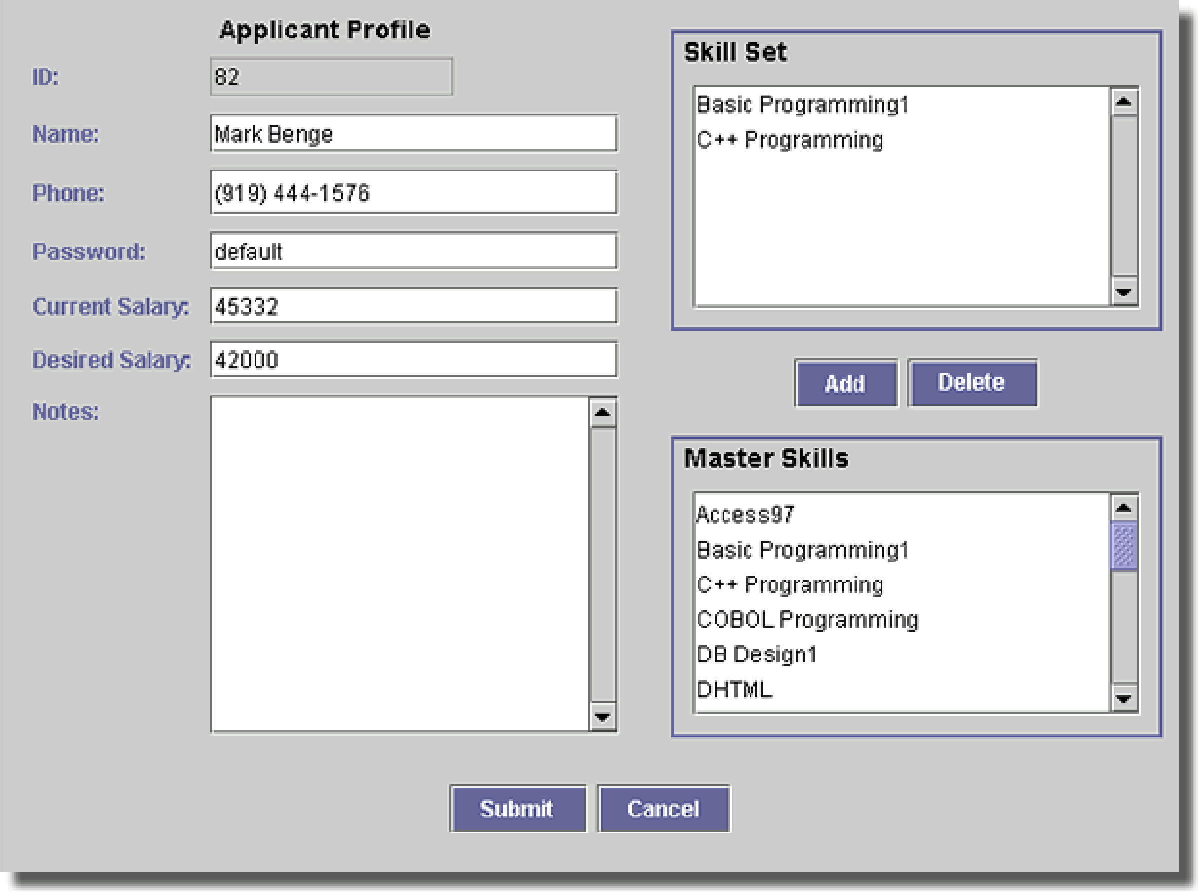
○ Open ● Under Offer ○ Closed

Notes:

This is a sample screenshot of the applet.

Applicants upon logging in are shown positions that he/she is qualified for. If they are interested in a specific position they can select the ‘Request Interview’ option.

Users can also modify their profile online.



Applicant Profile

ID:	82
Name:	Mark Benge
Phone:	(919) 444-1576
Password:	default
Current Salary:	45332
Desired Salary:	42000
Notes:	

Skill Set

- Basic Programming1
- C++ Programming

Master Skills

- Access97
- Basic Programming1
- C++ Programming
- COBOL Programming
- DB Design1
- DHTML

The user has the option to modify their information or a new user can enter their information and add their skills through this screen.

(Below) Code for the login screen of the applet. The method retrieves the applicant based on the entered ID and password.

```
/**
 * Perform the retrieveApplicant method.
 * @param UserID int
 * @param password java.lang.String
 */
public void retrieveApplicant(int UserID, char[] password) throws com.ibm.ivj.eab.dab.DAException, InvalidLoginException{
        //create String object to hold characters in array
        String newPass = new String("");

        //use for loop to concatenate characters to end of String
        for(int i = 0; i<password.length; i++)
        {
                newPass = newPass + password[i];
        }

        //query db for applicant passing an userid and password
        getDbApplicantManager().select(" where ID = " + UserID + " and PASSWORD = '" + newPass + "'");
        java.util.Vector applicant = getDbApplicantManager().items().getVector();

        //check vector size - if empty throw exception
        if (applicant.size() == 0){
                throw new InvalidLoginException("Invalid UserID or Password, please try again.");
        }
        else{
        //if applicant is there, create new applicantData object and set ListModel's applicantData object
                jobsdb.Applicant dbApp = (jobsdb.Applicant)applicant.get(0);
                ApplicantData app = new ApplicantData();
                app.setApplicantCurrentSalary(dbApp.getCurrentsalary());
                app.setApplicantDesiredSalary(dbApp.getDesiredsalary());
                app.setApplicantId(dbApp.getId());
                app.setApplicantName(dbApp.getName());
                app.setApplicantNotes(dbApp.getNotes());
                app.setApplicantPassword(dbApp.getPassword());
                app.setApplicantPhone(dbApp.getPhone());

                setApplicant(app);

        }
}
```

```
/**
 * retrieves all companies from database and adds it to the list model.
 *
 * @exception DAException If data access exception occurs when accessing database.
 */
public void retrieveallItems() throws DAException {

    //retrieve all the Companies in the db, sorted by name
    getDbCompanyManager().select("Order By Name");

    //using getVector method place all items in database into vector skills
    java.util.Vector companies = getDbCompanyManager().items().getVector();

    //loop through the vector, taking each skill and creating a new object reference to insert
    //into the JList
    for (int i = 0; i < companies.size(); i++)
    {
        CompanyData companyX = new CompanyData();
        setDbCompany((jobsdb.Company) companies.elementAt(i));
        companyX.setCompanyId(getDbCompany().getId());
        companyX.setCompanyName(getDbCompany().getName());
        companyX.setCompanyContact(getDbCompany().getContact());
        companyX.setCompanyEnrollDate(getDbCompany().getEnroll());
        companyX.setCompanyPhone(getDbCompany().getPhone());
        companyX.setCompanyNotes(getDbCompany().getNotes());

        //add element to list and calls toString method
        addElement(companyX);
    }

}
```

(Left)
The retrieveAllItems method obtains all companies from the database and populates the corresponding list box.

(Below)
The saveItem method validates the skill name for 0 or over 20 characters. It also determines based on the ID whether to save a new skill or update a selected skill.

```
/**
 * Perform the saveItem method.
 * this method checks to see if save is an add or an update
 * @param obj java.lang.Object
 * @param index int
 */
public void saveItem(Object obj, int index) throws com.ibm.ivj.eab.dab.DAException, MissingInformationException,
                    StringTooLongException, DuplicatesException{

    //cast obj argument to a SkillData object named saveSkill
    SkillData saveSkill = (SkillData) obj;

    //check if skill name is missing
    if (saveSkill.getSkillName().length() == 0)
    {
        throw new MissingInformationException("Please enter a skill name.");
    }
    //check length of skill name
    else if (saveSkill.getSkillName().length() > 20)
    {
        throw new StringTooLongException("Skill names cannot have more than 20 letters.");
    }


    //check SkillId - if 0 then call insertItem, else updateItem
    if (saveSkill.getSkillId() == 0) {
        for (int i = 0; i < this.size(); i++) {
            //check for duplicate names
            if (saveSkill.getSkillName().equalsIgnoreCase(this.getElementAt(i).toString())) {
            throw new DuplicatesException( "The Skill " + saveSkill.getSkillName() + " is already in the
            database.");
            }
        }
        insertItem(saveSkill);
    }
    else {
        updateItem(saveSkill, index);
    }

  }
```

Web Presence

Project Description

- ❑ Obtaining requirements from clients through interviews with domain experts, white papers, etc.
- ❑ Developing use cases based on requirements.
- ❑ Creation of a website for a fictional car broker carConnexions using FrontPage 2000. Incorporate VBScript to develop client-side validation, calculations and make website more interactive.
- ❑ Developing use cases and an Entity-Relationship Diagram (ERD) based on client's requirements for database application using Microsoft Access 2000.
- ❑ Designing a user-interface using Visual Basic for Applications (VBA) to aid in entry of data into database which included a financial calculator. Incorporated ActiveX components, SQL queries, reports, Data Access Pages. Allowing clients/users to access server database over the Internet.

USE CASE: Maintain Dealer Information

Actors - eCar Connexion staff

Purpose - To allow eCar Connexion staff to add/edit/delete dealer information.

Overview - The eCar Connexion staff can add/edit/delete all the dealer information. The information includes the Dealer Name, Contact Name, Dealer Address, Business Phone Number, Dealer Region, Email Address, Web Site URL, and Sell/Lease capabilities. The Web Site URL is optional.

Typical Course of Events

1. This use case begins when the eCar Connexion staff enters the Dealer Information Section.
2. System Response -Add, Edit, and Delete option displays.
3. Staff selects option

If Add option selected, see section Add Dealer Information.

If Edit option selected, see section Edit Dealer Information.

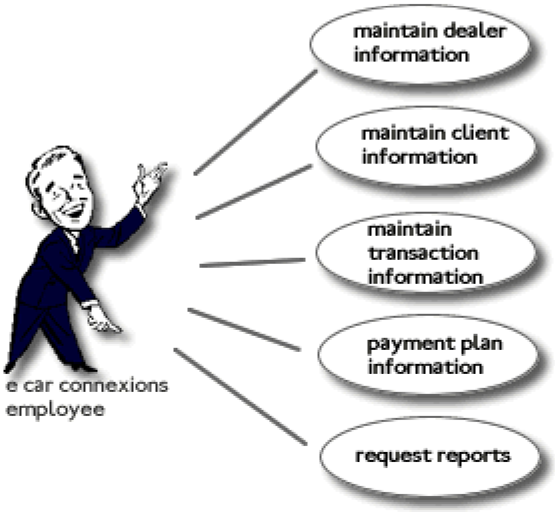
If Delete option selected, see section Delete Dealer Information.

Section - Add Dealer Information

1. System Response - An empty dealer form displays
2. The staff enters their information.
3. The staff saves their information.
4. System Response - Dealer Information added.

Alternative Scenario

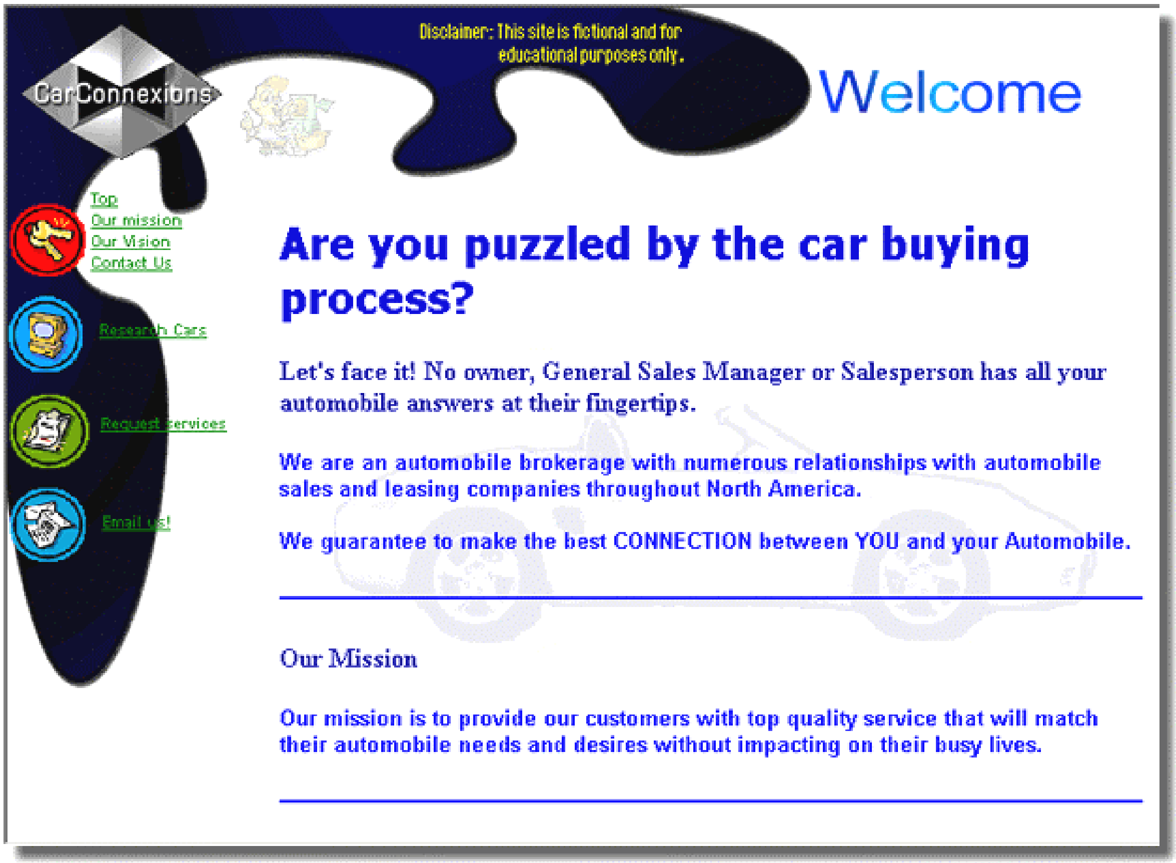
- Line 3 - Required item missing. Indicate error and allow re-entry at Line 2.
- Line 3 - Required item in invalid format. Indicate error and allow re-entry at Line 2.


maintain dealer information
maintain client information
maintain transaction information
payment plan information
request reports
e car connexions employee

The project began with obtaining requirements from the client to develop use cases.

(Left) A sample use case from our database application for users maintaining dealer information.

(Above) The use case diagram for the application. Employee is able to maintain client, dealer, and transaction information, calculate payments or request reports.



Disclaimer: This site is fictional and for educational purposes only.

Welcome

Are you puzzled by the car buying process?

Let's face it! No owner, General Sales Manager or Salesperson has all your automobile answers at their fingertips.

We are an automobile brokerage with numerous relationships with automobile sales and leasing companies throughout North America.

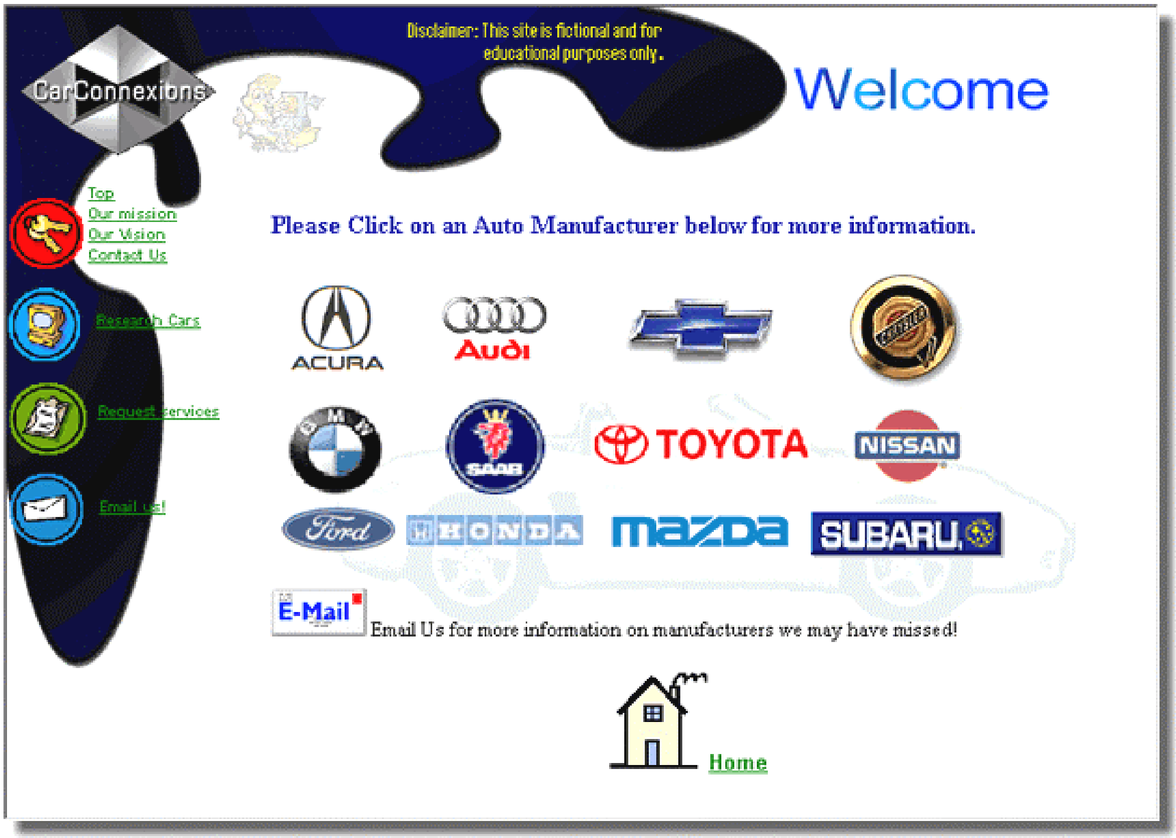
We guarantee to make the best CONNECTION between YOU and your Automobile.

Our Mission

Our mission is to provide our customers with top quality service that will match their automobile needs and desires without impacting on their busy lives.

This is the main page of the website. We decided to use a Puzzle theme for the website and designed the look with the theme in mind.

This "Puzzle" theme was created using Fireworks and cut into two pieces then laid out between two frames.



Disclaimer: This site is fictional and for educational purposes only.

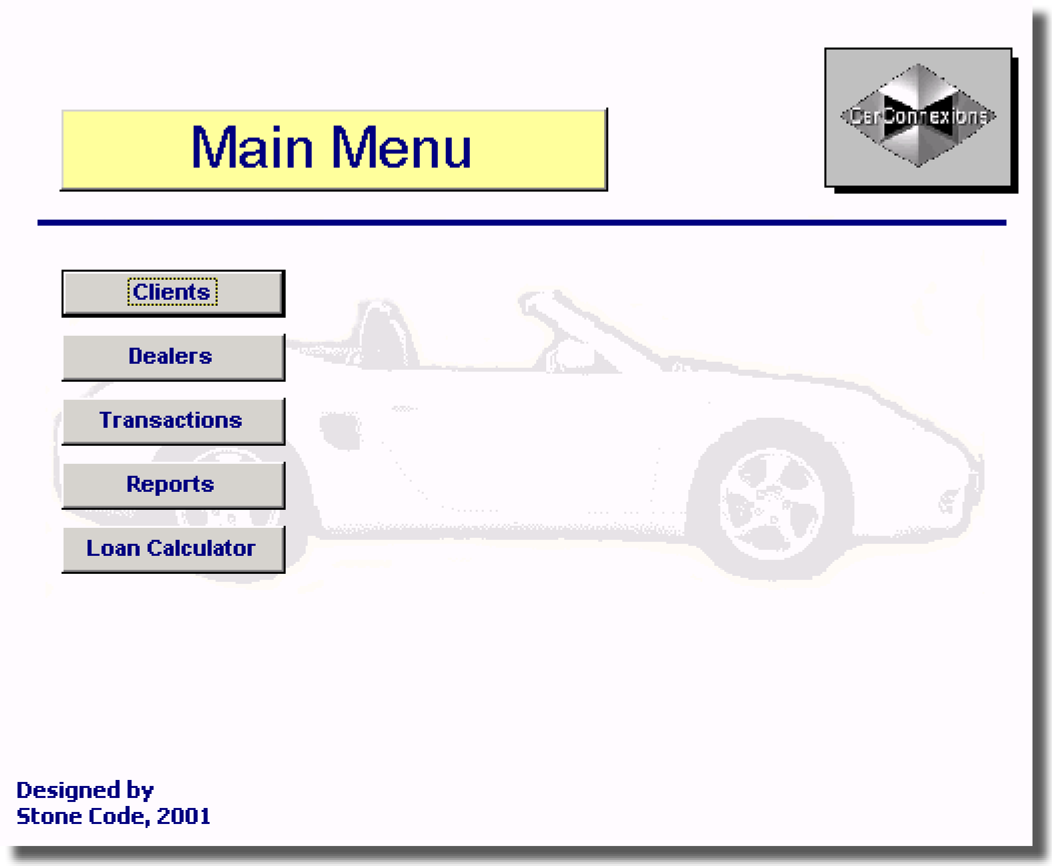
Welcome

Please Click on an Auto Manufacturer below for more information.

Email Us for more information on manufacturers we may have missed!

Users have the option of researching cars before requesting the service of the company.

Clicking on a Manufacturer logo opens a new window showing the manufacturer's website.



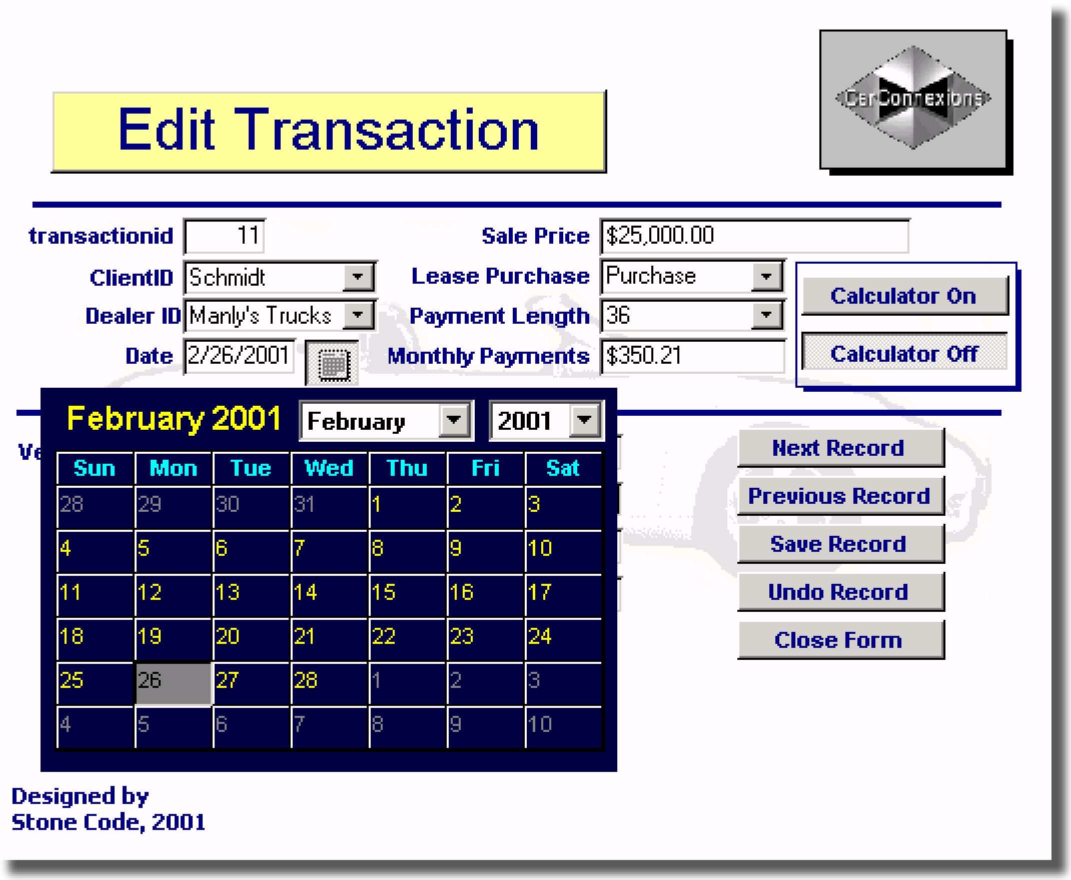
Main Menu

Designed by
Stone Code, 2001

Database Main menu. Developed a user interface using VBA.

Utilized the background and similar color scheme as the website.

This along with the utilization of switchboard menus facilitate ease-of-use through familiarity and navigability.



Edit Transaction

transactionid	11	Sale Price	$25,000.00
ClientID	Schmidt	Lease Purchase	Purchase
Dealer ID	Manly's Trucks	Payment Length	36
Date	2/26/2001	Monthly Payments	$350.21

February 2001 February 2001

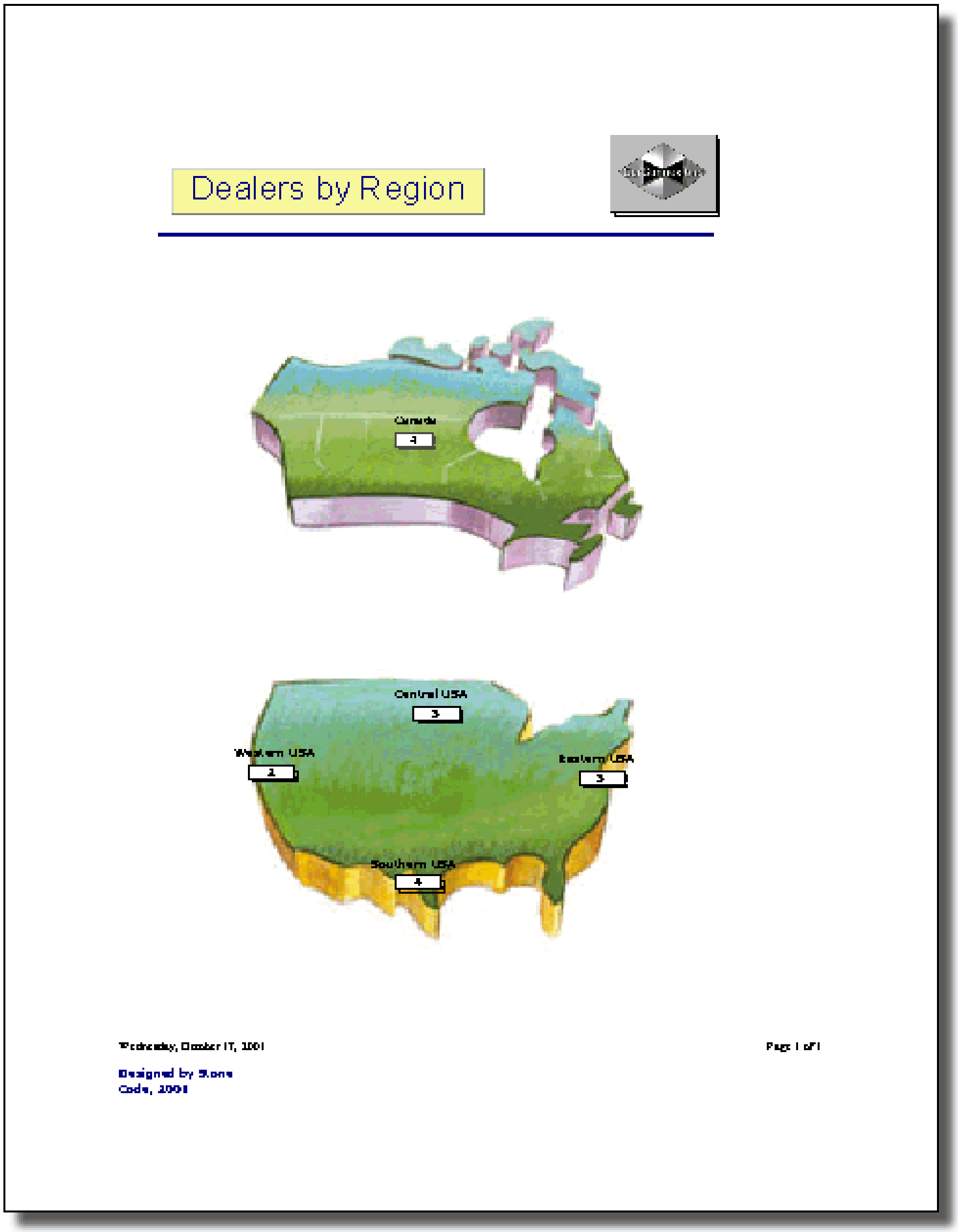
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	1	2	3
4	5	6	7	8	9	10

Designed by
Stone Code, 2001

Edit Transaction menu.

Utilized a drop down Active X Calendar to aid in proper date entry.

A toggle button to open the loan calculator is available to the user.



Dealers by Region

Wednesday, October 17, 2001

Page 1 of 1

Designed by Stone Code, 2001

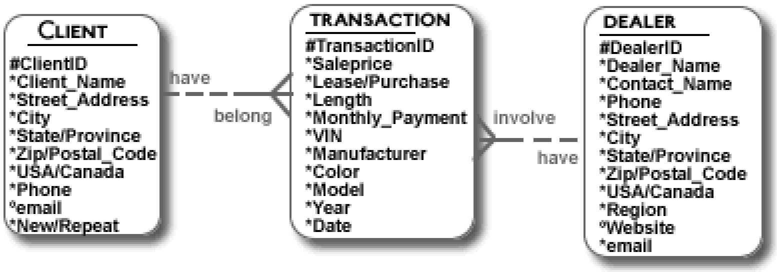
This is a report displaying the number of Dealers by Region.

Queried the database for the information but decided on a more interesting means of presenting the data.

Again, stressing the consistency of design.

Superimposed query data onto graphical representations of Canada and the US.

An Entity Relationship Diagram (ERD) was developed before building the database. Outlined are the three entities, their attributes, and relationships.



CLIENT
#ClientID
*Client_Name
*Street_Address
*City
*State/Province
*Zip/Postal_Code
*USA/Canada
*Phone
°email
*New/Repeat

TRANSACTION
#TransactionID
*Saleprice
*Lease/Purchase
*Length
*Monthly_Payment
*VIN
*Manufacturer
*Color
*Model
*Year
*Date

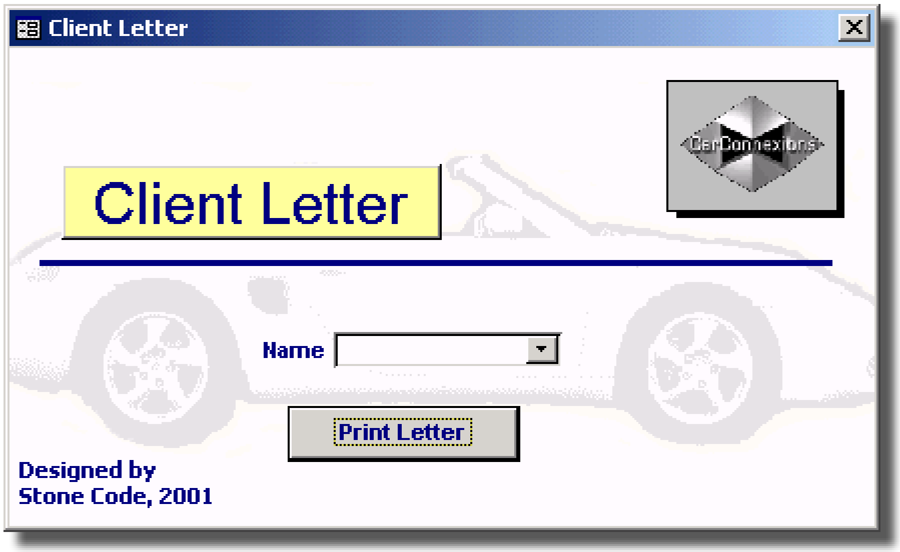
DEALER
#DealerID
*Dealer_Name
*Contact_Name
*Phone
*Street_Address
*City
*State/Province
*Zip/Postal_Code
*USA/Canada
*Region
°Website
*email

- Each client may have one or more transactions.

- Each transaction must belong to one and only one client.

- Each dealer may have one or more transactions

- Each transaction must involve one and only one dealer

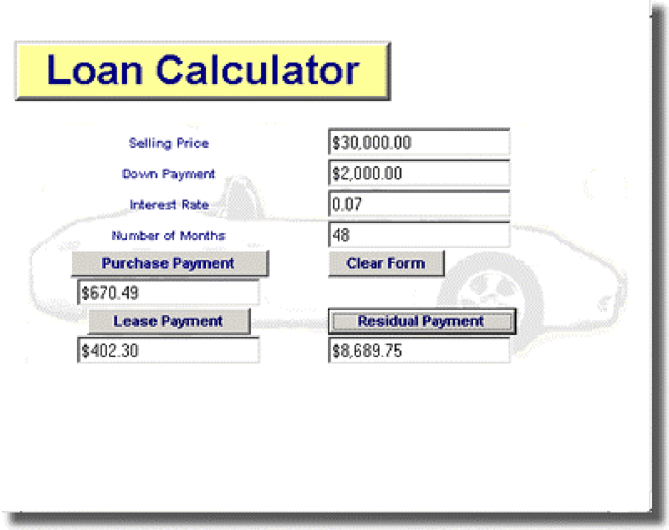


The user has the option of selecting a client and printing a pre-written thank-you letter in Microsoft Word.

Below is the code behind this form.

```
Option Explicit

Private Sub cmdPrint_Click()
'this sub on click opens a Thank you word document, placing the string in a
'placeholder for name and prints the document.

Dim objWord As Object
Dim strName As String

'validate that there is something in the combobox. If there isn't anything,
'prompts for a Client Name.
If IsNull(Me.cboName.Value) Then
    MsgBox "Please select a Client", , "Invalid Client."
    Me.cboName.SetFocus
    Exit Sub
Else
'places combobox value in strName
    strName = Me.cboName.Value
'create application object
    Set objWord = CreateObject("Word.Application")
'open template called thankyou2.doc
    objWord.Documents.Open "d:\My Documents\Thankyou2.doc"

    With objWord
'make application visible
    .Visible = True
'find bookmark called "Name" and select it
    .ActiveDocument.Bookmarks("Name").Select
'with bookmark selected, change it to strName Value
    .Selection.Text = strName
    End With
'Print letter in foreground so that it won't close before finishing
    objWord.ActiveDocument.PrintOut Background:=False
'close document without saving changes
    objWord.ActiveDocument.Close savechanges:=False
'quit word application
    objWord.Quit
End If
'reset wordobject
Set objWord = Nothing
End Sub
```

```
<script language = "vbscript">
<!--
Option Explicit
        Dim SP, DP, IR, MLP, MPP, RP, months
Function MonthlyLP(SP, DP, IR, Months)
        'this calculates the monthly lease payment
        MonthlyLP = 0.6 * (((SP - DP) * (IR/12))/(1 - (1 +(IR/12))^(-months)))
End Function

Sub cmdLease_onclick()
'this sub on click checks each entry for validity.  If validity is  assured, then function
‘MLP is called with each argument passed
‘and placed into variable MLP, which is then shown in the msgbox.
        If validation = false then
           exit sub
        Else
           MLP = MonthlyLP(SP, DP, IR, months)
           frmcalculator.txtleasepayment.value = formatCurrency(MLP)
        End If
End sub

Function Validation()
'this function checks for valid values in txt boxes
        SP = frmcalculator.txtsellingprice.value
        DP = frmcalculator.txtdownpayment.value
        IR = frmcalculator.txtinterestrate.value
        months = frmcalculator.txtmonths.value

' If a txtbox is not numeric or 0 then a message box asking for a valid
' entry is shown.  Focus is set on the needed txtbox.
        If IsNumeric(SP) = False or SP = "0" Then
          MsgBox "Please enter a valid number in Selling Price."
          frmcalculator.txtsellingprice.focus
          validation = False
          Exit function
        ElseIf IsNumeric(DP) = False or DP = "0" Then
          MsgBox "Please enter a valid number in Down Payment."
          frmcalculator.txtdownpayment.focus
          validation = False
          Exit Function
        ElseIf IsNumeric(IR) = False Then
          MsgBox "Please enter a valid number in Interest Rate."
          frmcalculator.txtInterestRate.focus
          validation = False
          Exit Function
        ElseIf IsNumeric(Months) = False or Months = "0" Then
          MsgBox "Please enter a valid number of months."
          frmcalculator.txtmonths.focus
          validation = False
          Exit Function
        Else
          Validation = True
        End If
' Check to see if IR is "0".  If so, then "0" becomes "0.000000001" to calculate
' the Payments with an interest rate as if it were "0" yet not break the divisible
' by zero rule.
        If IR = "0" Then
                IR = "0.000000001"
        Else

                If frmcalculator.txtinterestrate.value > 1 Then
                  frmcalculator.txtinterestrate.value = frmcalculator.txtinterestrate.value/100
                ElseIf frmcalculator.txtinterestrate.value < 0 Then
                  MsgBox "Please enter a valid Interest Rate."
                  frmcalculator.txtinterestrate.value = "0"
                  frmcalculator.txtinterestrate.value.focus
                  exit function
                Else
                  frmcalculator.txtinterestrate.value = frmcalculator.txtinterestrate.value
                End If
                IR = frmcalculator.txtinterestrate.value
                frmcalculator.txtinterestrate.value = frmcalculator.txtinterestrate.value
        End if
End Function
-->
</script>
```



Selling Price $30,000.00
Down Payment $2,000.00
Interest Rate 0.07
Number of Months 48
$670.49
$402.30
$8,689.75

A loan calculator was available in both the application and the intranet designed by our team. Utilized VBScript to add the functionality required to this page.

Here is the VBScript code behind calculating the lease payment.